

AD-A066 156

NORTHROP RESEARCH AND TECHNOLOGY CENTER PALOS VERDES --ETC F/G 20/5  
THEORETICAL MODELING OF MOLECULAR AND ELECTRON KINETIC PROCESSES--ETC(U)  
JAN 79 W B LACINA

UNCLASSIFIED

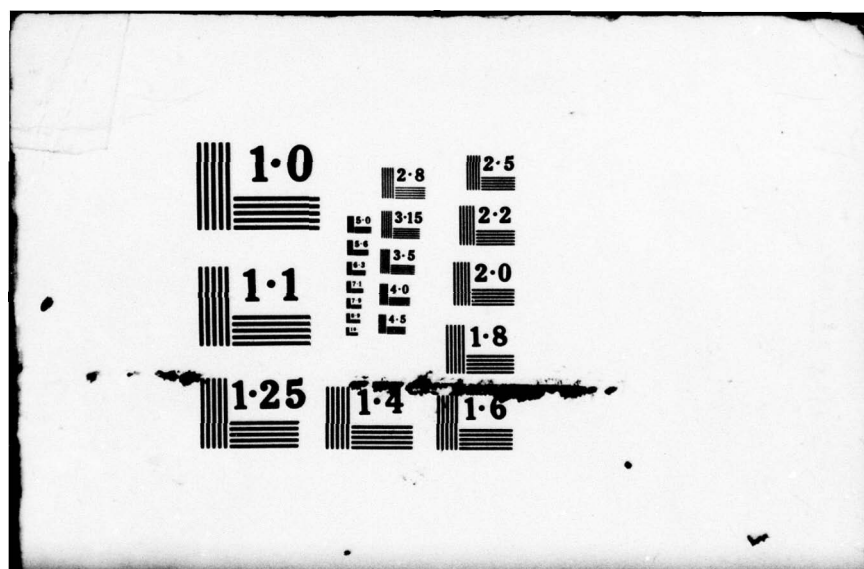
NRTC-79-TR-VOL-2

N00014-78-C-0499

NL

1 OF 2  
ADA  
066156







**LEVEL**

12  
B.S.

NRTC-79-7R

AD A0 661 56

THEORETICAL MODELING OF MOLECULAR AND ELECTRON  
KINETIC PROCESSES

Volume II

FORTTRAN Computer Program Listings  
Generalized Laser Kinetics Synthesis and Analysis  
Boltzmann Electron Kinetics Analysis

January, 1979

William B. Lacina



DDC FILE COPY

Sponsored by

OFFICE OF NAVAL RESEARCH  
Department of the Navy

Contract No. N00014-78-C-0499

This document has been approved  
for public release and sale; its  
distribution is unlimited.

**NORTHROP**

Research and Technology Center

79 03 21 030

⑥ THEORETICAL MODELING OF MOLECULAR AND ELECTRON  
KINETIC PROCESSES.

Volume II.

FORTRAN Computer Program Listings,  
Generalized Laser Kinetics Synthesis and Analysis  
Boltzmann Electron Kinetics Analysis.

⑪ January, 1979

⑨ Final technical rept. 12 Jun 78-15 Jan 79,

⑩ William B./Lacina

⑭ NRTC-79-7R-VOL-2

NORTHROP RESEARCH AND TECHNOLOGY CENTER  
One Research Park  
Palos Verdes Peninsula, California 90274  
Telephone: (213) 377-4811

⑫ 131 p.

Sponsored by

OFFICE OF NAVAL RESEARCH  
Department of the Navy

Contract No. N00014-78-C-0499

⑮

407 696

alt

ACCESSION for	
NTIS	Section <input checked="" type="checkbox"/>
DOC	B.R. Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	Avail. and/or SPECIAL
A	



## UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THEORETICAL MODELING OF MOLECULAR AND ELECTRON KINETIC PROCESSES. Vol. II: Fortran Listings, General Laser Kinetics Synthesis and Analysis; Boltzmann Kinetics Analysis		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 12 June 1978 to 15 Jan. 1979
7. AUTHOR(s) AD-A066155 William B. Lacina		6. PERFORMING ORG. REPORT NUMBER NRTC-79-7R
9. PERFORMING ORGANIZATION NAME AND ADDRESS NORTHROP Research and Technology Center One Research Park Palos Verdes Peninsula, Ca. 90274		8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0499
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Dept. of the Navy 800 North Quincy Street Arlington, Virginia 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Attn.: 613C:MAK
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research, Pasadena, California		12. REPORT DATE January, 1979
		13. NUMBER OF PAGES 126
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Unlimited.		
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <b>DISTRIBUTION STATEMENT A</b>            Approved for public release;            Distribution Unlimited         </div>		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Refer to Vol. I of this report.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Vol. I of this report describes and documents a comprehensive and reasonably general computer analysis applicable to a broad class of transient, electrically excited laser systems. The theoretical model is based on a coupled analysis of molecular kinetics, electron kinetics, external driving circuit, and radiative extraction. A complete discussion has been given in Vol. I. Vol. II contains the FORTRAN listings for these programs and their subroutines, which were developed for the CDC 6000 and CYBER series computer system.		

THEORETICAL MODELING OF MOLECULAR AND  
ELECTRON KINETIC PROCESSES

Volume I, II

Contract Number:	N00014-78-C-0499 613C:MAK
Principal Investigator:	Dr. William B. Lacina (213) 377-4811, Ext. 322
Name of Contractor:	NORTHROP Corporation Northrop Research and Technology Center One Research Park Palos Verdes Peninsula California 90274
Scientific Officer:	Director, Physics Program Physical Sciences Division Office of Naval Research Department of the Navy 800 N. Quincy Street Arlington, Virginia 22217
Technical Monitor:	Dr. Robert Behringer Office of Naval Research Pasadena, California
Contract Period:	12 June 1978 - 15 Jan. 1979

Reproduction of this document in whole or in part is permitted  
for any purpose of the United States Government.

The views and conclusions contained in this document are those  
of the author and should not be interpreted as necessarily repre-  
senting the official policies, either expressed or implied, of the  
Office of Naval Research or of the United States Government.



THEORETICAL MODELING OF MOLECULAR AND  
ELECTRON KINETIC PROCESSES

Volume II

FORTRAN Computer Program Listings  
Generalized Laser Kinetics Synthesis and Analysis  
Boltzmann Electron Kinetics Analysis

TABLE OF CONTENTS

---

Program LASER .....	1
Subroutines:	
1) SYNTH .....	32
2) ANALYZE .....	54
3) DECODE .....	56
4) UPDATE .....	57
5) PLASMA .....	61
6) SHAPE .....	64
7) BOLTZ.....	65
8) SIMEQ .....	83
9) GEAR .....	84
10) PLOT .....	96
11) AXIS .....	104
12) INTERP .....	105
13) SIMPSON .....	105
14) EDITOR .....	106
15) COVER .....	107
16) HEADINX .....	108
Program ELECT .....	111

---

```

PROGRAM LASER (INPUT,OUTPUT,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6=OUTPUT, LASER 2
1 TAPE7,TAPE8,TAPE9,TAPE10=INPUT) LASER 3
C LASER 4
C ----- LASER 5
C THIS PROGRAM IS A GENERAL LASER KINETICS SYNTHESIS CODE WHICH CAN LASER 6
C BE USED TO AUTOMATICALLY GENERATE SUBROUTINES REQUIRED FOR A LASER 7
C COUPLED ANALYSIS OF MOLECULAR AND ELECTRON KINETICS, OPTICAL EX- LASER 8
C TRACTION, AND EXTERNAL DRIVING CIRCUIT. THE MOLECULAR KINETICS LASER 9
C MECHANISMS ARE DEFINED BY AN ARBITRARILY LONG INPUT QUEUE OF SYM- LASER 10
C BOLIC REACTIONS, FOLLOWED BY THEIR FORWARD AND/OR REVERSE RATE LASER 11
C CONSTANTS, WHICH ARE AUTOMATICALLY TRANSLATED INTO COMPUTER-CODED LASER 12
C EQUATIONS. THE SYNTHESIZED SUBROUTINES REQUIRED FOR THE COMPLETE- LASER 13
C LY COUPLED ANALYSIS ARE COMPILED AND ADDED TO ALL OTHER REQUIRED LASER 14
C SUBROUTINES FOR EXECUTION. A MULTISTEP GEAR INTEGRATION SCHEME LASER 15
C FOR STIFF SYSTEMS OF DIFFERENTIAL EQUATIONS IS USED FOR THE FINAL LASER 16
C ANALYSIS. THE TREATMENT OF ELECTRON KINETICS CONSISTS OF NUMERI- LASER 17
C CAL SOLUTION OF THE BOLTZMANN TRANSPORT EQUATION, WITH SUPERELAS- LASER 18
C TIC COLLISIONS AND ELECTRON-ELECTRON SCATTERING INCLUDED. THE LASER 19
C RADIATIVE EXTRACTION IS FORMULATED IN TERMS OF THE CAVITY PHOTON LASER 20
C DENSITY ( $N = I/C \cdot h \cdot \nu$ ), WITH A SPONTANEOUS EMISSION SOURCE TERM LASER 21
C AND AMPLIFICATION FROM STIMULATED EMISSION. LASER 22
C LASER 23
C THIS CODE WAS DEVELOPED BY -- LASER 24
C LASER 25
C ----- LASER 26
C I I LASER 27
C I DR. WILLIAM B. LACINA I LASER 28
C I NORTHROP RESEARCH AND TECHNOLOGY I LASER 29
C I ONE RESEARCH PARK I LASER 30
C I PALOS VERDES PENINSULA, CA 90274 I LASER 31
C I TEL: (213) 377-4811, EXT. 322 I LASER 32
C I I LASER 33
C ----- LASER 34
C LASER 35
C LASER 36
C LASER 37
C LASER 38
C COMPLETE DOCUMENTATION OF THE PRESENT CODE (INCLUDING A DISCUSSION LASER 39
C OF THE MATHEMATICAL FORMULATION, DESCRIPTION OF THE SUBROUTINES LASER 40
C AND NUMERICAL TECHNIQUES, AND INSTRUCTIONS FOR USAGE) IS AVAILABLE LASER 41
C IN PUBLISHED REPORTS (1978). LASER 42
C ----- LASER 43
C LASER 44
C LASER 45
C DECLARATION STATEMENTS -- LASER 46
C LASER 47
C LASER 48
C DIMENSION KF(200), KR(200), GAS(200), RATEK(200), FLAG(200), LASER 49
1 LEV1(25), LEV2(25), N1(25), N2(25), VSIG(2,25), POWER(25), U(25), LASER
2 NEL(25), PROCESS(4,25), Q(501,25), QM(501), QMOM(501,2), F(501), LASER
3 S(501), G(501), A(501,3), B(501), EV(501), NAME(30), MASS(30), LASER
4 FI(30), E(30), PLOTS(30), PMAX(30), NTIME(30), RPCT(30), POP(32), LASER
5 NO(32), NDOT(32), DNYDTN(32,8), SCRTCH(32,12), YMAX(32), ERR(32), LASER
6 PHI(32,65), NOUT(201,2,32), TIME(201), ISUS(201,2), COND(201,2), LASER
7 IBEAM(201), RAD(201,3), ALPHA(201,4), TB(21), JB(21), COMMENT(5), LASER
8 YO(2), DY(2), RATE(2), RNAME(10), IMAGE(8), TITLE(3), KAPTION(4), LASER
9 IO(10), OUT(10), DUM(5,2), LABEL(5,2), LINE(250), ERROR(10), LASER
$ FATAL(10), MOM(50), ITAU(30) LASER 58

```



C	REAL NO, N1, N2, NDOT, KF, KR, NTOT, NOUT, NMOL, MASS, MOLWT, KB,	LASER	59
	1 LDIDT, KTE, MU, ISUS, NE, IBEAM, INDUCT, KVOLT, LOSS, LENGTH,	LASER	60
	2 JBEAM, JSUS, KVCN, IONIZE, JB	LASER	61
C	INTEGER GAS, TYPE, TITLE, RATE, LHS, RHS, TODAY, GENDATE, WARN,	LASER	62
	1 WORD	LASER	63
C	LOGICAL CONVRGE, ERROR, REJECT, STOP, FATAL, LIST, OUT, ELECT,	LASER	64
	1 EXPAND, TEST, MISSING, OUTSIDE, STIM, END, MODIFY, ILLEGAL,	LASER	65
	2 ARC, FLAG, INTRP, PLOTS, REPEAT, ERRORS, FE	LASER	66
C	EQUIVALENCE (B.QM)	LASER	67
C	-----	LASER	68
C	LABELED COMMON BLOCKS --	LASER	69
C	COMMON / DATA / RATEK, KF, KR, VSIG, E	LASER	70
C	COMMON / DISCH / INDUCT, CAPAC, RESIST, MU, AREA, DIST	LASER	71
C	COMMON / GAINS / GNET, GAMMA, GAIN, ABSORB, OMEGA, LENGTH, CAVITY	LASER	72
C	COMMON / SOURCE / UPLUS, JBEAM, DVDX, FACTOR, ENERGY, SB, S0	LASER	73
C	COMMON / CONST / NTOT, TMOL, FREQ, HNU	LASER	74
C	COMMON / TIMES / TR, TF, TFALL, TC, TB, JB, INTRP, NPTS, UNITS	LASER	75
C	-----	LASER	76
C	DATA INITIALIZATION --	LASER	77
C	DATA MESH, ITMAX, METHOD, MAXDER, IDEG, IO, NSIZE, LIMIT, NCYCLE,	LASER	78
	1 KOUNT / 500, 100, 1, 7, 2, 10*1, 10, 2*200, 20 /	LASER	79
C	DATA EMAX, TMAX, EPS, ETA, RE, ELIMIT, PCT, PER, WARN / 20.0,	LASER	80
	1 50.0, 3*0.001, 30.0, 2*5.0, 8*WARNING: /	LASER	81
C	DATA KB, EE, P1, EFMT, FFMT / 1.30E-23, 1.602E-19, 3.14159,	LASER	82
	1 9H(1PE10.2), 7H(F10.2) /	LASER	83
C	DATA OUT, EXPAND, REPEAT, FATAL, FE, ERROR, ERRORS, PLOTS, FLAG,	LASER	84
	1 LIST / 64*.FALSE., 201*.TRUE. /	LASER	85
C	-----	LASER	86
C	NAMelist PARAMETERS --	LASER	87
C	NAMelist / CONTROL / EMAX, MESH, EXPAND, ITMAX, TMAX, EPS, IDEG,	LASER	88
	1 PCT, KOUNT, NCYCLE, LIMIT, MAXDER, METHOD, ETA, FATAL, IO	LASER	89
C	NAMelist / PARAM / TPULSE, TMOL, TE, PTOT, ATM	LASER	90
C	NAMelist / RATES / KF, KR	LASER	91
C		LASER	92

	NAMelist / EBEAM / JBEAM, ENERGY, FACTOR, TR, TF, TC, TFALL, UA,	LASER	116
	1 UB, JB, TB, UNITS	LASER	117
C		LASER	118
	NAMelist / CIRCUIT / KVOLT, INDUCT, CAPAC, RESIST, AREA, DIST	LASER	119
C		LASER	120
	NAMelist / OPTICAL / LOSS, REFLECT, GAMMA, AREA, LENGTH, CAVITY,	LASER	121
	1 OMEGA	LASER	122
C		LASER	123
C	-----	LASER	124
C		LASER	125
C	ARRAY STORAGE IS DEFINED BY THE FOLLOWING DIMENSION DECLARATORS --	LASER	126
C		LASER	127
	MAX = 200	LASER	128
	M1 = MAX+1	LASER	129
C	(MAX = MAXIMUM VALUE OF NCYCLE, THE NUMBER OF INTERVALS INTO WHICH	LASER	130
C	THE TOTAL PULSE DURATION IS SUBDIVIDED FOR BOLTZMANN CALCULATIONS	LASER	131
C	AND OUTPUT GENERATION.)	LASER	132
	MGRID = 500	LASER	133
C	(MGRID = MAXIMUM NUMBER OF BINS INTO WHICH THE ELECTRON ENERGY	LASER	134
C	RANGE MAY BE PARTITIONED IN THE E- KINETICS ANALYSIS.)	LASER	135
	KMAX = 200	LASER	136
C	(KMAX = MAXIMUM NUMBER OF REACTIONS)	LASER	137
	NMAX = 30	LASER	138
C	(NMAX = MAXIMUM VALUE OF NTYPE, THE NUMBER OF SPECIES)	LASER	139
	NMAXP2 = NMAX+2	LASER	140
C	(NMAXP2 IS MAXIMUM NUMBER OF EQUATIONS FOR SPECIES AND CIRCUIT.)	LASER	141
	NKMAX = 25	LASER	142
C	(NKMAX = MAXIMUM VALUE OF NK = NUMBER OF SECONDARY E- REACTIONS)	LASER	143
C		LASER	144
C	IF IT IS DESIRED TO CHANGE DIMENSION STORAGE, THE ABOVE DECLARATOR	LASER	145
C	VALUES MUST BE MODIFIED, AND THE FOLLOWING MISCELLANEOUS ARRAYS	LASER	146
C	MUST BE CHANGED (AS SHOWN) IN THE DIMENSION STATEMENT --	LASER	147
C		LASER	148
C	KF(KMAX), KR(KMAX), GAS(KMAX), LEV1(NKMAX), LEV2(NKMAX), N1(NKMAX)	LASER	149
C	N2(NKMAX), VSIG(2*NKMAX), POWER(NKMAX), U(NKMAX), NEL(NKMAX),	LASER	150
C	PROCESS(4,NKMAX), Q(MGRID+1,NKMAX), QM(MGRID+1), QMOM(MGRID+1,2),	LASER	151
C	F(MGRID+1), G(MGRID+1), A(MGRID+1,3), B(MGRID+1), EV(MGRID+1),	LASER	152
C	FI(NMAX), NAME(NMAX), MASS(NMAX), E(NMAX), PLOTS(NMAX), PMAX(NMAX)	LASER	153
C	NTIME(NMAX), RPCT(NMAX), NO(NMAXP2), POP(NMAXP2), NDOT(NMAXP2),	LASER	154
C	DNYDTN(NMAXP2,8), NOUT(MAX+1,2,NMAX+2), YMAX(NMAXP2), ERR(NMAXP2),	LASER	155
C	SCRATCH(NMAXP2,12), PHI(NMAX+2,2*NMAX+5), S(MGRID+1), TIME(MAX+1),	LASER	156
C	ISUS(MAX+1,2), COND(MAX+1,2), IBEAM(MAX+1), RAD(MAX+1,3),	LASER	157
C	ALPHA(MAX+1,4), RATEK(KMAX)	LASER	158
C		LASER	159
C	NOTE ALSO THAT DIMENSIONS OF ARRAYS WHICH OCCUR IN LABELLED COMMON	LASER	160
C	BLOCKS MUST BE ACCORDINGLY MODIFIED IN ANY SUBROUTINE WHERE THEY	LASER	161
C	OCCUR. IN PARTICULAR, COMMON / DATA / CONTAINS DIMENSIONS IN THE	LASER	162
C	SYNTHESIZED SUBROUTINES *JACOB* AND *DNOT*.	LASER	163
C		LASER	164
C	IF CERTAIN DIMENSION DECLARATORS DO NOT AGREE WITH THOSE ON AN IN-	LASER	165
C	PUT DATA FILE (NSCRATCH) ACCESSED DURING EXECUTION, AN EXIT OCCURS.	LASER	166
C		LASER	167
C	-----	LASER	168
C		LASER	169
C	FILE USAGE --	LASER	170
C		LASER	171
C	INPUT CARD FILE --	LASER	172



C	KARDS = 10	LASER	173
C		LASER	174
C	SUBROUTINE *EDITOR*, WHICH EDITS THE INPUT CARD IMAGES, CREATES AN	LASER	175
C	EFFECTIVE INPUT FILE ON TAPE 5, WHICH IS THEN SUBSEQUENTLY USED.	LASER	176
C		LASER	177
C	INPUT E- CROSS SECTION FILE --	LASER	178
C	INPUT = 8	LASER	179
C	REWIND INPUT	LASER	180
C		LASER	181
C	FILE OF UPDATED E- CROSS SECTIONS --	LASER	182
C	NDATA = 9	LASER	183
C	REWIND NDATA	LASER	184
C		LASER	185
C	SCRATCH TAPE USED IN PROGRAM SETUP --	LASER	186
C	NSCRTCH = 4	LASER	187
C		LASER	188
C	FILE FOR SYNTHESIS OF RATE SUBROUTINES --	LASER	189
C	NTAPE = 7	LASER	190
C	REWIND NTAPE	LASER	191
C	MTAPE = 3	LASER	192
C	REWIND MTAPE	LASER	193
C	LTAPE = 2	LASER	194
C	REWIND LTAPE	LASER	195
C		LASER	196
C	-----	LASER	197
C		LASER	198
C	DO 7 NK = 1, NKMAX	LASER	199
C	7 LEV1(NK) = LEV2(NK) = 0	LASER	200
C	DO 1 I = 1, NMAXP2	LASER	201
C	YMAX(I) = 1.0	LASER	202
C	DO 1 N = 1, 8	LASER	203
C	1 DNYDTN(I, N) = 0.	LASER	204
C		LASER	205
C	C IS JUST SOME ARBITRARY CONSTANT USED TO TAG RATES WHICH ARE NOT	LASER	206
C	EXPLICITLY INITIALIZED LATER BY INPUT --	LASER	207
C	C = -SQRT(1.25963782)	LASER	208
C	DO 62 K = 1, KMAX	LASER	209
C	62 KF(K) = KR(K) = C	LASER	210
C		LASER	211
C	KB = KB/EE	LASER	212
C	-----	LASER	213
C	CALL DATE (TODAY)	LASER	214
C	-----	LASER	215
C		LASER	216
C	REWIND NSCRTCH	LASER	217
C	READ (NSCRTCH) TITLE, GENDATE	LASER	218
C	IF (EOF(NSCRTCH)) 4,3	LASER	219
C		LASER	220
C	-----	LASER	221
C	SYNTHESIS OF MOLECULAR KINETICS SUBROUTINES	LASER	222
C	-----	LASER	223
C		LASER	224
C	SYNTHESIZE SUBROUTINE *DNDT* AND *JACOB* TO CALCULATE THE VALUES	LASER	225
C	OF DNI/DT AND D/DNJ(DNI/DT) AS A FUNCTION OF TIME, AND SUBROUTINE	LASER	226
C	*LEVELS* TO DEFINE THE CORRESPONDENCE OF LOWER AND UPPER STATES IN	LASER	227
C	THE E- KINETICS ANALYSIS WITH THE MOLECULAR SPECIES LABELS --	LASER	228
C		LASER	229

C	READ DATA BLOCK NO. 1 --	LASER	230
C	4 CALL EDITOR (KARDS,LIST)	LASER	231
C	GENDATE = TODAY	LASER	232
C	CALL SYNTH (LTAPE, MTAPE, NTAPE, NSCRTCH, INPUT, NSIZE, NMAX, GAS,	LASER	233
C	1 KMAX, NKMAX, LEV1, LEV2, TODAY)	LASER	234
C	AT THE TERMINATION OF PROGRAM GENERATION, THE SYNTHESIZED SUBROU-	LASER	235
C	TINES (80 BCD CHARACTER RECORDS) ARE STORED ON FILE *MTAPE* = 3,	LASER	236
C	WHICH IS USED AS A SOURCE FILE TO BE COMPILED (AND CATALOGED AS A	LASER	237
C	BCD UPDATE FILE, IF DESIRED). NUMERICAL DATA ASSOCIATED WITH RATE	LASER	238
C	CONSTANTS AND INFORMATION CHARACTERIZING THE REACTION SCHEME EN-	LASER	239
C	COUNTERED (UPON WHICH THE SYNTHESIZED SUBROUTINES ARE BASED) IS	LASER	240
C	STORED ON FILE *NSCRTCH* = 4. IF FUTURE EXECUTION OF THE ANALYSIS	LASER	241
C	IS INTENDED, *NSCRTCH* MUST ALSO BE CATALOGUED AND SAVED.	LASER	242
C	CALL EXIT	LASER	243
C	-----	LASER	244
C	GENERATE AN UPDATED ELECTRON CROSS SECTION FILE --	LASER	245
C	READ DATA BLOCK NO. 2 --	LASER	246
C	3 CALL EDITOR (KARDS,LIST)	LASER	247
C	CALL UPDATE (INPUT, NDATA, NTAPE, .NOT,LIST, TODAY)	LASER	248
C	-----	LASER	249
C	CALL COVER (TITLE,2)	LASER	250
C	-----	LASER	251
C	READ DATA BLOCK NO. 3 --	LASER	252
C	CALL EDITOR (KARDS,.NOT,LIST)	LASER	253
C	-----	LASER	254
C	GENERATE COMMENT CARD INFORMATION, IF ANY --	LASER	255
C	-----	LASER	256
C	LC = 0	LASER	257
C	55 READ (5,101) IMAGE	LASER	258
C	IF (EOF(5)) 56,59	LASER	259
C	59 IF (LC,GT,0) GO TO 57	LASER	260
C	WRITE (6,304)	LASER	261
C	WRITE (6,302)	LASER	262
C	57 LC = LC+1	LASER	263
C	WRITE (6,303) IMAGE	LASER	264
C	IF (LC,NE,30) GO TO 55	LASER	265
C	WRITE (6,302)	LASER	266
C	LC = 0	LASER	267
C	GO TO 55	LASER	268
C	56 IF (LC,NE,0) WRITE (6,302)	LASER	269
C	READ DATA BLOCK NO. 4 --	LASER	270
C	CALL EDITOR (KARDS,LIST)	LASER	271
C	-----	LASER	272
C	READ GENERAL CONTROL PARAMETERS THAT DEFINE ACCURACIES, SCOPE OF	LASER	273
C	CALCULATIONS, ITERATION LIMITS, INTERPOLATION ORDER, METHOD OF	LASER	274
C	INTEGRATION, AND OUTPUT OPTIONS --	LASER	275
C	-----	LASER	276
C		LASER	277
C		LASER	278
C		LASER	279
C		LASER	280
C		LASER	281
C		LASER	282
C		LASER	283
C		LASER	284
C		LASER	285
C		LASER	286

C		LASER	287
C	-----	LASER	288
	READ (5,CONTROL)	LASER	289
C	-----	LASER	290
C		LASER	291
	IF (METHOD.NE.0.AND.METHOD.NE.2) METHOD = 1	LASER	292
	IF (MAXDER.GT.8) MAXDER = 8	LASER	293
	IF (MAXDER.EQ.8.AND.METHOD.NE.0) MAXDER = 7	LASER	294
	IF (NCYCLE.LE.0) NCYCLE = 1	LASER	295
	IF (NCYCLE.GT.MAX) NCYCLE = MAX	LASER	296
	IF (LIMIT.LT.0) LIMIT = 0	LASER	297
	IF (LIMIT.GT.NCYCLE) LIMIT = NCYCLE	LASER	298
C		LASER	299
	READ (NSCRTCH) MAXGAS, NTYPE, MAXK, KTYPE, MAXNK, NK, ERRORS	LASER	300
C		LASER	301
	ERROR(1) = ERRORS	LASER	302
	ERROR(8) = (MAXGAS.NE.NMAX).OR.(MAXNK.NE.NKMAX).OR.(MAXK.NE.KMAX)	LASER	303
C		LASER	304
C	IF THE FOLLOWING CONDITION OCCURS, DIMENSION STORAGE IS INADEQUATE	LASER	305
C		LASER	306
	ERROR(9) = (NTYPE.GT.NMAX).OR.(KTYPE.GT.KMAX).OR.(NK.GT.NKMAX)	LASER	307
C		LASER	308
C	TRUNCATE DATA FILE *NSCRTCH* IF NECESSARY --	LASER	309
	IF (NTYPE.GT.NMAX) NTYPE = NMAX	LASER	310
	IF (KTYPE.GT.KMAX) KTYPE = KMAX	LASER	311
	IF (NK.GT.NKMAX) NK = NKMAX	LASER	312
C		LASER	313
	READ (NSCRTCH) (GAS(N), N = 1,NTYPE)	LASER	314
	READ (NSCRTCH) (LEV1(N), LEV2(N), N = 1,NK)	LASER	315
C		LASER	316
	DO 72 I = 1,10	LASER	317
	72 OUT(I) = IO(I).NE.0	LASER	318
C		LASER	319
C	-----	LASER	320
C	READ GENERAL EXPERIMENTAL PARAMETERS RELATING TO TEMPERATURE,	LASER	321
C	PRESSURE, PULSE LENGTH, AND MODIFICATION OF RATE CONSTANTS --	LASER	322
C	-----	LASER	323
C		LASER	324
	PTOT = ATM = TPULSE = TE = 0.	LASER	325
	TMOL = 300.	LASER	326
C		LASER	327
C	-----	LASER	328
	READ (5,PARAM)	LASER	329
C	-----	LASER	330
C		LASER	331
	IF (MESH.GT.MGRID) MESH = MGRID	LASER	332
	MESHPI = MESH+1	LASER	333
	IF (TMOL.LE.0.) TMOL = 300.	LASER	334
	IF (TE.LE.0.) TE = TMOL	LASER	335
	IF (PTOT.EQ.0.) PTOT = 760.*ATM	LASER	336
	ATM = PTOT/760.	LASER	337
	UNIT = 1.0	LASER	338
	IF (TPULSE.LE.0.) GO TO 42	LASER	339
	TT = TPULSE	LASER	340
43	IF (TT.GT.1) GO TO 42	LASER	341
	TT = 1000.*TT	LASER	342
	UNIT = UNIT/1000.	LASER	343



	GO TO 43	LASER	344
42	TOUT = TPULSE/NCYCLE	LASER	345
	DTIME = TOUT/UNIT	LASER	346
C		LASER	347
C	-----	LASER	348
C	READ OPTICAL RESONATOR PARAMETERS (REFLECTIVITY, LOSS, LENGTH,	LASER	349
C	OMEGA, ETC.) --	LASER	350
C	-----	LASER	351
C		LASER	352
	LOSS = REFLECT = GAMMA = 0.	LASER	353
	LENGTH = CAVITY = AREA = OMEGA = 0.	LASER	354
C		LASER	355
C	-----	LASER	356
C	READ (5,OPTICAL)	LASER	357
C	-----	LASER	358
C		LASER	359
	IF (LENGTH.LE.0.) LENGTH = 0.01	LASER	360
	IF (CAVITY.LE.0.) CAVITY = LENGTH	LASER	361
	IF (CAVITY.LT.LENGTH) CAVITY = LENGTH	LASER	362
	STIM = REFLECT.GT.0.	LASER	363
	IF (REFLECT.LE.0.) REFLECT = 1.E-20	LASER	364
	IF (LOSS.LT.0.) LOSS = 0.	LASER	365
	R = REFLECT/100.	LASER	366
	LOSS = LOSS/100.	LASER	367
	IF (GAMMA.GT.0.) LOSS = 0.	LASER	368
	IF (OMEGA.LE.0.) OMEGA = AREA/CAVITY**2	LASER	369
	IF (GAMMA.EQ.0.) GAMMA = (LOSS - 0.5*ALOG(R))/LENGTH	LASER	370
	TCAVITY = CAVITY/(30.*LENGTH*GAMMA)	LASER	371
	PASS = 100.*LOSS	LASER	372
	OMEGA4P = OMEGA/4./PI	LASER	373
C		LASER	374
C	-----	LASER	375
C	READ EXPERIMENTAL ELECTRICAL AND CIRCUIT PARAMETERS --	LASER	376
C	-----	LASER	377
C		LASER	378
	TR = 0.	LASER	379
	TF = TC = 10000.	LASER	380
	TFALL = 1.0	LASER	381
	DO 517 I = 1,21	LASER	382
517	TB(I) = JB(I) = 0.	LASER	383
	UNITS = 1.0E-09	LASER	384
	JBEAM = ENERGY = 0.	LASER	385
	FACTOR = 1.0	LASER	386
	SU = DU = 0.	LASER	387
	UA = UB = 0.	LASER	388
	DEPOSIT = 0.	LASER	389
C		LASER	390
C	-----	LASER	391
C	READ (5,EBeam)	LASER	392
C	-----	LASER	393
		LASER	394
	IF (UB.GT.EMAX) UB = EMAX	LASER	395
C	FOR THE SQUARE WAVE S(U) = 1, UA ≤ U ≤ UB, THE AVERAGE ENERGY	LASER	396
C	UPLUS = <U> IS GIVEN BY --	LASER	397
	UPLUS = (UA + UB)/2.	LASER	398
	SMAX = JB(1)	LASER	399
	IF (SMAX.LT.0.) GO TO 519	LASER	400

NPTS = 1	LASER	401
T0 = TB(1)	LASER	402
DO 515 I = 2,21	LASER	403
T1 = TB(1)	LASER	404
IF (T1.LE.T0) GO TO 518	LASER	405
NPTS = NPTS+1	LASER	406
SI = JB(1)	LASER	407
IF (SI.LT.0.) GO TO 519	LASER	408
IF (SI.GT.SMAX) SMAX = SI	LASER	409
515 T0 = T1	LASER	410
518 INTRP = NPTS.GT.1	LASER	411
C COMPUTE NORMALIZED E-BEAM CURRENT DENSITY SHAPE FUNCTION --	LASER	412
DO 521 I = 1,NPTS	LASER	413
521 JB(I) = JB(I)/SMAX	LASER	414
GO TO 520	LASER	415
519 INTRP = .FALSE.	LASER	416
C	LASER	417
520 KVOLT = 0.	LASER	418
PDISCH = 0.	LASER	419
AREA = DIST = 1.	LASER	420
INDUCT = RESIST = 0.	LASER	421
CAPAC = 1.0	LASER	422
C	LASER	423
C -----	LASER	424
READ (5,CIRCUIT)	LASER	425
C -----	LASER	426
C	LASER	427
VOLT = 1000.*KVOLT	LASER	428
ELECT = KVOLT.NE.0.	LASER	429
IF (DIST.EQ.0.) DIST = 1.0	LASER	430
C	LASER	431
ELECT = ELECT.AND.NK.NE.0	LASER	432
NEQ = NTYPE	LASER	433
IF (ELECT) NEQ = NEQ+2	LASER	434
NP1 = NTYPE+1	LASER	435
NP2 = NTYPE+2	LASER	436
C	LASER	437
C -----	LASER	438
C READ UPDATED RATES FOR ALL PROCESSES --	LASER	439
C -----	LASER	440
C	LASER	441
C READ NUMERICAL RATE DATA PROVIDED AT THE TIME OF PROGRAM SYNTHESIS	LASER	442
C (STORED ON TAPE NSCRTCH), AND (POSSIBLY) MODIFIED BY INPUT FROM	LASER	443
C THE SRATES ... \$ CARD. NOTE THAT CERTAIN RATES ARE ABSENT IF THEY	LASER	444
C DID NOT APPEAR AT PROGRAM SYNTHESIS. THESE RATES ARE NOT ACCESS-	LASER	445
C SIBLE BY INPUT, AND ARE IGNORED IF AN ATTEMPT IS MADE TO SPECIFY	LASER	446
C THEM ON THE SRATES ... \$ CARD. HOWEVER, FOR SECONDARY ELECTRON	LASER	447
C COLLISIONS (FOR WHICH RATES ARE NORMALLY OBTAINED BY DEFAULT TO	LASER	448
C E- KINETICS CALCULATIONS), FIXED INPUT VALUES FOR RATES MAY BE	LASER	449
C ASSIGNED BY THE SRATES ... \$ CARD IF THERE IS NO ELECTRIC FIELD	LASER	450
C (I.E., EVCM = 0 SPECIFIED ON THE FOREGOING SCIRCUIT ... \$ CARD).	LASER	451
C	LASER	452
C -----	LASER	453
C READ (5,RATES)	LASER	454
C -----	LASER	455
C	LASER	456
C REWIND MTAPE	LASER	457

STOP = .FALSE.	LASER	458
K0 = 24	LASER	459
IF (KOUNT.LE.0) KOUNT = K0	LASER	460
IF (KOUNT.GT.K0) KOUNT = K0	LASER	461
NSKIP = K0 - KOUNT	LASER	462
IF (NSKIP.LE.0) NSKIP = 1	LASER	463
ENCODE (10,109,PAGE) NSKIP	LASER	464
L = LC = K = N = 0	LASER	465
ILLEGAL = MODIFY = .FALSE.	LASER	466
71 READ (INSCRTCH) LSUM, LSUM, LABEL, RATE, FK, RK, RNAME, COMMENT	LASER	467
IF (EOF(INSCRTCH)) 70,91	LASER	468
91 NFLAG = 1H	LASER	469
TEST = .FALSE.	LASER	470
K = K+1	LASER	471
DO 84 I = 1,NTYPE	LASER	472
NL = NR = 0	LASER	473
DO 503 J = 1,5	LASER	474
IF (LABEL(J,1).EQ.1) NL = NL+1	LASER	475
503 IF (LABEL(J,2).EQ.1) NR = NR+1	LASER	476
NI = NR-NL	LASER	477
IF (I.EQ.1.AND.NL.EQ.1.AND.NR.EQ.1) NI = 1	LASER	478
84 NTIME(I) = NI	LASER	479
IMAGE(1) = RATE(1)	LASER	480
IMAGE(2) = RATE(2)	LASER	481
DECODE (1,100,RATE(1)) R1	LASER	482
DECODE (1,100,RATE(2)) R2	LASER	483
IF (R2.EQ.1HX) IMAGE(2) = 1H	LASER	484
IF (R1.NE.1HV.AND.R2.NE.1HV) GO TO 73	LASER	485
N = N+1	LASER	486
VSIG(1,N) = VSIG(2,N) = 0.	LASER	487
C	LASER	488
C	LASER	489
C	LASER	490
COUNT NET NUMBER OF ELECTRONS (RHS-LHS) --	LASER	491
NEL(N) = 0	LASER	492
DO 92 M = 1,5	LASER	493
IF (LABEL(M,1).EQ.2) NEL(N) = NEL(N)-1	LASER	494
92 IF (LABEL(M,2).EQ.2) NEL(N) = NEL(N)+1	LASER	495
C	LASER	496
IF (.NOT.ELECT) GO TO 74	LASER	497
IF (R1.EQ.1HV) IMAGE(1) = 1H	LASER	498
IF (R2.EQ.1HV) IMAGE(2) = 1H	LASER	499
GO TO 73	LASER	500
C	LASER	501
C	LASER	502
C	LASER	503
C	LASER	504
(IF EVCM = 0. AND IF KF OR KR ARE NOT SPECIFIED FOR A SECONDARY ELECTRON PROCESS. IT IS ASSUMED BY THE PROGRAM THAT THERE WAS AN IMPLIED INPUT OF KF = 0 AND/OR KR = 0 FOR THAT PROCESS.)	LASER	505
74 IF (R1.NE.1HV) GO TO 76	LASER	506
IF (KF(K).EQ.C) KF(K) = 0.	LASER	507
VSIG(1,N) = KF(K)	LASER	508
76 IF (R2.NE.1HV) GO TO 73	LASER	509
IF (KR(K).EQ.C) KR(K) = 0.	LASER	510
VSIG(2,N) = KR(K)	LASER	511
73 IF (KF(K).EQ.C) GO TO 67	LASER	512
IF (IMAGE(1).NE.1H) GO TO 67	LASER	513
KF(K) = C	LASER	514
L = L+1		



J = 1HF	LASER	515
NFLAG = 2H**	LASER	516
TEST = ILLEGAL = .TRUE.	LASER	517
ENCODE (10,305,LINE(L)) J, K	LASER	518
STOP = .TRUE.	LASER	519
67 IF (KR(K).EQ.C) GO TO 68	LASER	520
IF (IMAGE(2).NE.1H ) GO TO 68	LASER	521
KR(K) = C	LASER	522
L = L+1	LASER	523
J = 1HR	LASER	524
NFLAG = 2H**	LASER	525
TEST = ILLEGAL = .TRUE.	LASER	526
ENCODE (10,305,LINE(L)) J, K	LASER	527
STOP = .TRUE.	LASER	528
C	LASER	529
C	LASER	530
C	LASER	531
C	LASER	532
C	LASER	533
C	LASER	534
AT THIS POINT, WITH THE EXCEPTION OF CHANGES ENCOUNTERED AND PER-	LASER	535
MITTED BY INPUT, VECTORS KF AND KR CONTAIN ALL C-VALUES, AND	LASER	536
LINE(L) CONTAINS NAMES OF RATES INACCESSIBLE BY INPUT FOR THE PRO-	LASER	537
GRAM EXECUTION.	LASER	538
68 IF (KF(K).EQ.C) GO TO 63	LASER	539
ENCODE (50,323,COMMENT)	LASER	540
MODIFY = .TRUE.	LASER	541
IF (.NOT.TEST) NFLAG = 2H *	LASER	542
ENCODE (10,301,RATE(1)) KF(K)	LASER	543
IF (KF(K).EQ.0.) RATE(1) = 1H	LASER	544
GO TO 64	LASER	545
63 KF(K) = FK	LASER	546
64 IF (KR(K).EQ.C) GO TO 69	LASER	547
ENCODE (50,323,COMMENT)	LASER	548
MODIFY = .TRUE.	LASER	549
IF (.NOT.TEST) NFLAG = 2H *	LASER	550
ENCODE (10,301,RATE(2)) KR(K)	LASER	551
IF (KR(K).EQ.0.) RATE(2) = 1H	LASER	552
GO TO 10	LASER	553
69 KR(K) = RK	LASER	554
10 WRITE (MTAPE) (NTIME(I), I = 1,NTYPE), RATE, RNAME	LASER	555
IF (.NOT.OUT(10)) GO TO 71	LASER	556
IF (LC.NE.0) GO TO 2	LASER	557
IF (K.EQ.1) GO TO 83	LASER	558
WRITE (6,102)	LASER	559
IF (MODIFY) WRITE (6,123)	LASER	560
IF (ILLEGAL) WRITE (6,124)	LASER	561
83 ILLEGAL = MODIFY = .FALSE.	LASER	562
WRITE (6,PAGE)	LASER	563
WRITE (6,103) GENDATE	LASER	564
IF (ELECT) WRITE (6,219)	LASER	565
WRITE (6,105)	LASER	566
2 LC = LC+1	LASER	567
IF (LC.EQ.KOUNT) LC = 0	LASER	568
WRITE (6,104) NFLAG, K, (RNAME(J), J = 1,5), RATE, COMMENT	LASER	569
GO TO 71	LASER	570
C	LASER	571
70 IF (.NOT.OUT(10)) GO TO 93		
WRITE (6,102)		
IF (MODIFY) WRITE (6,123)		
IF (ILLEGAL) WRITE (6,124)		

93 ERROR(7) = STOP	LASER	572
C	LASER	573
C	LASER	574
C	LASER	575
C	LASER	576
C	LASER	577
LINES = 6	LASER	578
WRITE (6,212)	LASER	579
C	LASER	580
IF (.NOT.ERROR(1)) GO TO 514	LASER	581
WORD = WARN	LASER	582
IF (FATAL(1)) WORD = 6HFATAL:	LASER	583
WRITE (6,201) WORD	LASER	584
LINES = LINES+5	LASER	585
C	LASER	586
514 IF (.NOT.ERROR(8)) GO TO 516	LASER	587
WORD = WARN	LASER	588
IF (FATAL(8)) WORD = 6HFATAL:	LASER	589
WRITE (6,208) WORD, MAXGAS, MAXK, MAXNK	LASER	590
LINES = LINES+5	LASER	591
C	LASER	592
516 IF (.NOT.ERROR(9)) GO TO 525	LASER	593
WORD = WARN	LASER	594
IF (FATAL(9)) WORD = 6HFATAL:	LASER	595
WRITE (6,209) WORD, NKMAX, KMAX, NMAX, NK, KTYPE, NTYPE	LASER	596
LINES = LINES+5	LASER	597
C	LASER	598
525 IF (.NOT.ERROR(7)) GO TO 75	LASER	599
WORD = WARN	LASER	600
IF (FATAL(7)) WORD = 6HFATAL:	LASER	601
WRITE (6,207) WORD	LASER	602
LINES = LINES + 5 + (L+7)/8	LASER	603
WRITE (6,108) (LINE(I), I = 1,L)	LASER	604
C	LASER	605
C	LASER	606
C	LASER	607
75 DO 5 I = 1,NTYPE	LASER	608
MASS(I) = 0.	LASER	609
5 E(I) = NO(I) = C	LASER	610
C	LASER	611
C	LASER	612
C	LASER	613
READ INPUT DATA: NAMES, CONCENTRATIONS, ENERGIES, MASSES --	LASER	614
L = NGAS = 0	LASER	615
PRESS = 0.	LASER	616
IONIZE = 1.0E-12	LASER	617
20 READ (5,101) TYPE, RATE, DUMMY, DUMMY, NPLOT	LASER	618
IF (EOF(5)) 11,31	LASER	619
31 BACKSPACE 5	LASER	620
READ (5,112) P0, E0, MOLWT	LASER	621
IF (TYPE.NE.3HRAD) GO TO 34	LASER	622
C	LASER	623
SPECIES *1* CORRESPONDS TO RADIATION --	LASER	624
NO(1) = P0	LASER	625
GO TO 20	LASER	626
C	LASER	627
34 IF (TYPE.NE.4HE(-)) GO TO 39	LASER	628
C	LASER	
SPECIES *2* CORRESPONDS TO ELECTRONS --		
NO(2) = P0		
PLOTS(2) = NPLOT.EQ.4HPLOT		



GO TO 20	LASER	629
39 IF (TYPE.NE.6HIONIZE) GO TO 81	LASER	630
IONIZE = P0	LASER	631
PLOTS(2) = NPLOT.EQ.4HPL0T	LASER	632
GO TO 20	LASER	633
81 PRESS = PRESS + P0	LASER	634
IF (P0.LE.0.) GO TO 58	LASER	635
NGAS = NGAS+1	LASER	636
NAME(NGAS) = TYPE	LASER	637
MASS(NGAS) = MOLWT	LASER	638
FI(NGAS) = P0	LASER	639
58 DO 8 I = 3,NTYPE	LASER	640
IF (TYPE.EQ.GAS(I)) GO TO 9	LASER	641
8 CONTINUE	LASER	642
L = L+1	LASER	643
LINE(L) = TYPE	LASER	644
ERROR(2) = .TRUE.	LASER	645
GO TO 20	LASER	646
9 IF (RATE(1).NE.1H ) NO(I) = P0*0.965E 19/TMOL	LASER	647
IF (RATE(2).NE.1H ) E(I) = E0	LASER	648
PLOTS(1) = NPLOT.EQ.4HPL0T	LASER	649
GO TO 20	LASER	650
C	LASER	651
11 IF (.NOT.ERROR(2)) GO TO 526	LASER	652
WORD = WARN	LASER	653
IF (FATAL(2)) WORD = 6HFATAL:	LASER	654
WRITE (6,202) WORD	LASER	655
LINES = LINES + 5 + (L+7)/8	LASER	656
WRITE (6,108) (LINE(I), I = 1,L)	LASER	657
C	LASER	658
526 DO 6 I = 1,NGAS	LASER	659
6 FI(I) = FI(I)/PRESS	LASER	660
IF (PTOT.EQ.0.) GO TO 27	LASER	661
DO 28 I = 3,NTYPE	LASER	662
28 IF (NO(I).NE.C) NO(I) = NO(I)*PTOT/PRESS	LASER	663
GO TO 26	LASER	664
27 PTOT = PRESS	LASER	665
26 NMOL = NTOT = 0.965E 19*PTOT/TMOL	LASER	666
IF (NO(2).LE.0.) NO(2) = IONIZE*NMOL	LASER	667
C	LASER	668
C CHECK INITIALIZATION OF POPULATION DENSITIES --	LASER	669
L = 0	LASER	670
DO 12 I = 1,NTYPE	LASER	671
IF (NO(I).NE.C) GO TO 12	LASER	672
ERROR(3) = .TRUE.	LASER	673
L = L+1	LASER	674
LINE(L) = GAS(I)	LASER	675
NO(I) = 0.	LASER	676
12 DNYDTN(1,1) = NO(1)	LASER	677
C	LASER	678
IF (.NOT.ERROR(3)) GO TO 527	LASER	679
WORD = WARN	LASER	680
IF (FATAL(3)) WORD = 6HFATAL:	LASER	681
WRITE (6,203) WORD	LASER	682
LINES = LINES + 5 + (L+7)/8	LASER	683
WRITE (6,108) (LINE(I), I = 1,L)	LASER	684
C	LASER	685

C	CHECK INITIALIZATION OF ENERGIES --	LASER	686
527	L = 0	LASER	687
	E(1) = E(2) = 0.	LASER	688
	DO 15 I = 3, NTYPE	LASER	689
	IF (E(I).NE.C) GO TO 15	LASER	690
	E(I) = 0.	LASER	691
	ERROR(4) = .TRUE.	LASER	692
	L = L+1	LASER	693
	LINE(L) = GAS(I)	LASER	694
	15 CONTINUE	LASER	695
C	IF (.NOT.ERROR(4)) GO TO 528	LASER	696
	WORD = WARN	LASER	697
	IF (FATAL(4)) WORD = 6HFATAL:	LASER	698
	WRITE (6,204) WORD	LASER	699
	LINES = LINES + 5 + (L+7)/8	LASER	700
	WRITE (6,108) (LINE(I), I = 1,L)	LASER	701
C	528 IF (.NOT.ELECT) GO TO 80	LASER	702
C	-----	LASER	703
C		LASER	704
C		LASER	705
C	IF THE BOLTZMANN ANALYSIS BELOW DOES NOT CONVERGE, CONTROL RETURNS	LASER	706
C	TO THIS POINT TO EXPAND THE ELECTRON ENERGY RANGE --	LASER	707
C		LASER	708
	TA = 0.	LASER	709
	IF (.NOT.REPEAT) GO TO 79	LASER	710
95	IF (EMAX.GE.5.) GO TO 87	LASER	711
	EMAX = EMAX*2.	LASER	712
	IF (EMAX.GT.5.) EMAX = 5.	LASER	713
	GO TO 89	LASER	714
87	EMAX = EMAX + 5.	LASER	715
89	IF (EMAX.LE.ELIMIT) GO TO 79	LASER	716
	WRITE (6,220) EMAX, ELIMIT	LASER	717
	GO TO 97	LASER	718
79	DE = EMAX/MESH	LASER	719
	E0 = 0.	LASER	720
	NA = UA/DE + 1	LASER	721
	NB = UB/DE + 1	LASER	722
	IF ((UA.NE.UB).AND.(NA.EQ.NB)) NB = NA+1	LASER	723
	UA = DE*(NA-1)	LASER	724
	UB = DE*(NB-1)	LASER	725
	DU = UB-UA	LASER	726
	SU = 0.	LASER	727
	IF (DU.GT.0.) SU = 1./DU	LASER	728
	DO 29 I = 1, MESHPI	LASER	729
	QMOM(I,1) = QMOM(I,2) = 0.	LASER	730
	EV(I) = E0	LASER	731
	S(I) = 0.	LASER	732
	IF (I.LT.NA.OR.I.GT.NB) GO TO 29	LASER	733
	S(I) = SU	LASER	734
29	E0 = E0 + DE	LASER	735
C		LASER	736
C	S(I) IS THE NORMALIZED SOURCE FUNCTION FOR SECONDARY ELECTRON	LASER	737
C	CREATION: INT(DU S(U)) = 1.	LASER	738
C		LASER	739
	REWIND NSCRCH	LASER	740
		LASER	741
		LASER	742

READ (NSCRTCH) TITLE	LASER	743
READ (NSCRTCH)	LASER	744
READ (NSCRTCH)	LASER	745
READ (NSCRTCH)	LASER	746
C -----	LASER	747
C	LASER	748
C	LASER	749
C IF THE SYNTHESIZED PROGRAM WAS CONSTRUCTED TO DEFAULT TO ELECTRON	LASER	750
C KINETICS CALCULATIONS AS THE SOURCE FOR SECONDARY ELECTRON COLLI-	LASER	751
C SION RATES, AND IF THE EXPERIMENTAL SITUATION CORRESPONDS TO AN	LASER	752
C ELECTRIC DISCHARGE (EVCN # 0), THE ELECTRON CROSS SECTION FILE IS	LASER	753
C PROCESSED --	LASER	754
C	LASER	755
CALL SECOND (TO)	LASER	756
IN = NTYPE	LASER	757
K = N = 0	LASER	758
J = 1	LASER	759
STOP = .FALSE.	LASER	760
14 READ (NSCRTCH) LHS, RHS, LABEL, RATE, FK, RK, RNAME	LASER	761
IF (EOF(NSCRTCH)) 60,505	LASER	762
505 DECODE (1,100,RATE(1)) R1	LASER	763
DECODE (1,100,RATE(2)) R2	LASER	764
IF (R1.NE.1HV.AND.R2.NE.1HV) GO TO 14	LASER	765
N = N+1	LASER	766
ENCODE (40,101,PROCESS(1,N)) (RNAME(L), L = 1,4)	LASER	767
C	LASER	768
C -----	LASER	769
C PROCESS THE INELASTIC ELECTRON CROSS SECTION FILE --	LASER	770
C -----	LASER	771
C	LASER	772
CALL PLASMA (NDATA, MGRID+1, MESH, LHS, RHS, RNAME, EV, F, G,	LASER	773
1 Q(1,N), U0, UM, IN, GAS, MISSING, REJECT, OUTSIDE, IDEG, OUT(8))	LASER	774
C	LASER	775
TEST = MISSING.OR.REJECT.OR.OUTSIDE	LASER	776
IF (.NOT.TEST) GO TO 32	LASER	777
STOP = STOP.OR.TEST	LASER	778
K = K+1	LASER	779
IF (J.GT.231) GO TO 33	LASER	780
ENCODE (50,322,LINE(J))	LASER	781
J = J+5	LASER	782
ENCODE (50,129,LINE(J)) K, (PROCESS(L,N), L = 1,4)	LASER	783
J = J+5	LASER	784
IF (MISSING) ENCODE (50,115,LINE(J))	LASER	785
IF (MISSING) J = J+5	LASER	786
IF (OUTSIDE) ENCODE (50,116,LINE(J)) EMAX	LASER	787
IF (OUTSIDE) J = J+5	LASER	788
IF (REJECT) ENCODE (50,117,LINE(J)) EMAX	LASER	789
IF (REJECT) J = J+5	LASER	790
IF (J.LE.225) GO TO 33	LASER	791
ENCODE (100,128,LINE(J))	LASER	792
J = J+10	LASER	793
GO TO 33	LASER	794
C	LASER	795
32 DO 24 L = 1,MESHP1	LASER	796
24 Q(L,N) = EV(L)*Q(L,N)	LASER	797
33 U(N) = 0.	LASER	798
L1 = LEV1(N)	LASER	799



IF (L1.EQ.0) GO TO 19	LASER	800
U(N) = - E(L1)	LASER	801
GO TO 13	LASER	802
19 DO 21 L = 1,5	LASER	803
I = LABEL(L,1)	LASER	804
IF (I.EQ.0) GO TO 13	LASER	805
21 U(N) = U(N)-E(I)	LASER	806
13 L2 = LEV2(N)	LASER	807
IF (L2.EQ.0) GO TO 23	LASER	808
U(N) = U(N) + E(L2)	LASER	809
GO TO 38	LASER	810
23 DO 22 L = 1,5	LASER	811
I = LABEL(L,2)	LASER	812
IF (I.EQ.0) GO TO 38	LASER	813
22 U(N) = U(N)+E(I)	LASER	814
38 CONTINUE	LASER	815
GO TO 14	LASER	816
C	LASER	817
60 ERROR(5) = STOP	LASER	818
FE = FATAL(5).AND.ERROR(5)	LASER	819
C	LASER	820
C	LASER	821
C	LASER	822
C	LASER	823
C	LASER	824
IN = NTYPE	LASER	825
TWOM = 2./1837.	LASER	826
STOP = .FALSE.	LASER	827
DO 17 I = 1,NGAS	LASER	828
IF (FI(I).EQ.0.) GO TO 17	LASER	829
FRACT = FI(I)	LASER	830
MISSING = REJECT = OUTSIDE = .FALSE.	LASER	831
ENCODE (40,106,RNAME) NAME(I)	LASER	832
C	LASER	833
ENCODE (50,120,IMAGE) NAME(I), NAME(I)	LASER	834
DECODE (50,100,IMAGE) (MOM(L), L = 1,50)	LASER	835
CALL DECODE (GAS, MOM, LHS, RHS, LABEL, DUM, 10, IN, 50)	LASER	836
C	LASER	837
CALL PLASMA (NDATA, MGRID,1, MESH, LHS, RHS, RNAME, EV, F, G, QM,	LASER	838
1 U0, UM, IN, GAS, MISSING, REJECT, OUTSIDE, IDEG, OUT(8))	LASER	839
C	LASER	840
TEST = MISSING.OR.REJECT.OR.OUTSIDE.OR.(MASS(I).LE.0.).OR.	LASER	841
(UM.LT.EMAX)	LASER	842
1 IF (.NOT.TEST) GO TO 36	LASER	843
K = K+1	LASER	844
IF (J.GT.221) GO TO 17	LASER	845
ENCODE (50,322,LINE(J))	LASER	846
J = J+5	LASER	847
ENCODE (50,129,LINE(J)) K, (RNAME(L), L = 1,4)	LASER	848
J = J+5	LASER	849
IF (MISSING) ENCODE (50,115,LINE(J))	LASER	850
IF (MISSING) J = J+5	LASER	851
IF (OUTSIDE) ENCODE (50,116,LINE(J)) EMAX	LASER	852
IF (OUTSIDE) J = J+5	LASER	853
IF (REJECT) ENCODE (50,117,LINE(J)) EMAX	LASER	854
IF (REJECT) J = J+5	LASER	855
IF (UM.LT.EMAX) ENCODE (50,118,LINE(J)) EMAX	LASER	856

IF (UM.LT.EMAX) J = J+5	LASER	857
IF (MASS(I).LE.0.) ENCODE (50,119,LINE(J))	LASER	858
IF (MASS(I).LE.0.) J = J+5	LASER	859
IF (J.LE.225) GO TO 17	LASER	860
ENCODE (100,128,LINE(J))	LASER	861
J = J+10	LASER	862
GO TO 17	LASER	863
C	LASER	864
C	LASER	865
C	LASER	866
C	LASER	867
C	LASER	868
C	LASER	869
36 DO 44 L = 1,MESHP1	LASER	870
FQ = FRACT*QM(L)	LASER	871
QMOM(L,1) = QMOM(L,1) + FQ	LASER	872
44 QMOM(L,2) = QMOM(L,2) + FQ/MASS(I)	LASER	873
C	LASER	874
17 STOP = STOP.OR.TEST	LASER	875
C	LASER	876
ERROR(6) = STOP	LASER	877
FE = FE.OR.(FATAL(6).AND.ERROR(6))	LASER	878
ERRORS = ERROR(5).OR.ERROR(6)	LASER	879
C	LASER	880
C	LASER	881
C	LASER	882
C	LASER	883
C	LASER	884
IF (.NOT.ERRORS) GO TO 98	LASER	885
WORD = WARN	LASER	886
IF (FE) WORD = 6HFATAL:	LASER	887
J = J-1	LASER	888
LINES = LINES + 5 + (J+4)/5	LASER	889
IF (REPEAT.OR.OUT(8).OR.OUT(9).OR.(LINES.GT.55))	LASER	890
1 WRITE (6,212)	LASER	891
LINES = 6	LASER	892
WRITE (6,205) WORD	LASER	893
WRITE (6,206) (LINE(L), L = 1,J)	LASER	894
C	LASER	895
98 FATAL(10) = .TRUE.	LASER	896
DO 46 L = 1,MESHP1	LASER	897
X = EV(L)	LASER	898
XSQ = TWOM*X*X	LASER	899
ERROR(10) = QMOM(L,1).LE.0.	LASER	900
IF (ERROR(10)) GO TO 47	LASER	901
A(L,1) = X/NMOL/QMOM(L,1)	LASER	902
46 A(L,2) = XSQ*NMOL*QMOM(L,2)	LASER	903
C	LASER	904
XBAR = DE/2.	LASER	905
DO 41 I = 1,MESH	LASER	906
CALL INTERP (2, XBAR, QMOM(I,1), EV, A(1,1), 1, MESHP1)	LASER	907
CALL INTERP (2, XBAR, QMOM(I,2), EV, A(1,2), 1, MESHP1)	LASER	908
41 XBAR = XBAR + DE	LASER	909
QMOM(MESHP1,1) = QMOM(MESH,1)	LASER	910
QMOM(MESHP1,2) = QMOM(MESH,2)	LASER	911
C	LASER	912
47 WORD = WARN	LASER	913

	IF (FATAL(10)) WORD = 6HFATAL:	LASER	914
	IF (ERROR(10)) WRITE (6,210) WORD	LASER	915
C	CALL SECOND (T1)	LASER	916
	TA = TA + (T1 - T0)	LASER	917
C	KTE = KB*TE	LASER	918
	EXPON = EXP(-DE/KTE)	LASER	919
	FB = 1.	LASER	920
	DO 37 I = 1,MESHP1	LASER	921
	F(I) = FB	LASER	922
37	FB = FB*EXPON	LASER	923
C		LASER	924
C	PROHIBIT FURTHER PLOTS OR TABULATIONS OF E- CROSS SECTION DATA --	LASER	925
C		LASER	926
C	OUT(8) = OUT(9) = .FALSE.	LASER	927
C		LASER	928
C	-----	LASER	929
C		LASER	930
C	TEST FOR ERROR CONDITIONS --	LASER	931
80	FE = ERRORS = .FALSE.	LASER	932
C		LASER	933
	DO 94 I = 1,10	LASER	934
	ERRORS = ERRORS.OR.ERROR(I)	LASER	935
	FE = FE.OR.(ERROR(I).AND.FATAL(I))	LASER	936
C	RESET ERROR FLAG --	LASER	937
94	ERROR(I) = .FALSE.	LASER	938
	IF (FE) WRITE (6,300)	LASER	939
	IF (FE) GO TO 99	LASER	940
C		LASER	941
	IF (REPEAT) GO TO 90	LASER	942
	IF (SU.EQ.0.) GO TO 45	LASER	943
C		LASER	944
C	PLOT OF NORMALIZED EXTERNAL IONIZATION SOURCE FUNCTION --	LASER	945
C		LASER	946
	WRITE (6,130) UPLUS	LASER	947
	Y0(1) = DY(1) = 0.	LASER	948
	CALL PLOT (1, MESH+1, 1, S, Y0, DY, EV, 0., 0., .TRUE., .TRUE.,	LASER	949
	1 .TRUE., .TRUE., .TRUE., TITLE, 1, 0)	LASER	950
	WRITE (6,131)	LASER	951
C		LASER	952
45	MU = 1000.	LASER	953
	TBOLTZ = NBOLTZ = 0	LASER	954
	VMAX = 0.9999*KVOLT	LASER	955
	NO(NP1) = CAPAC*VOLT	LASER	956
	NO(NP2) = 0.	LASER	957
	RD = DIST/AREA/(EE*NU)	LASER	958
	IF (INDUCT.EQ.0.) NO(NP2) = -VOLT/(RESIST + RD)	LASER	959
	ONLYDTN(NP1+1) = NO(NP1)	LASER	960
	ONLYDTN(NP2+1) = NO(NP2)	LASER	961
	HMIN = TOUT/1000.	LASER	962
	HMAX = TOUT	LASER	963
	H = HMIN	LASER	964
	DELTA = ETA	LASER	965
	JSTART = 0	LASER	966
	T = TP = NP = 0	LASER	967
	TI = TOUT	LASER	968
		LASER	969
		LASER	970



C	ENCODE (40,322,KAPTION)	LASER	971
C	-----	LASER	972
C	INTEGRATE EQUATIONS FROM T = 0 TO T = TPULSE --	LASER	973
C	-----	LASER	974
C		LASER	975
	30 END = NP.EQ.LIMIT	LASER	976
	HOUT = TOUT - TT	LASER	977
	IF (H,LT,HOUT) GO TO 18	LASER	978
C		LASER	979
C	BEFORE INTEGRATING FROM T = (T + H), WHICH WILL PASS THE CYCLE	LASER	980
C	TIME, WE PAUSE TO GENERATE OUTPUT EXTRAPOLATED TO TIME NP*TOUT --	LASER	981
C		LASER	982
	DO 16 I = 1,9	LASER	983
	OUT(I) = .FALSE.	LASER	984
	IF (IO(I).EQ.0) GO TO 16	LASER	985
	OUT(I) = NP.EQ.IO(I)*(NP/IO(I))	LASER	986
16	CONTINUE	LASER	987
	NP = NP+1	LASER	988
	TP = T + HOUT	LASER	989
	TIME(NP) = TP/UNIT	LASER	990
	IBEAM(NP) = JBEAM*SHAPE(TP)	LASER	991
	IF (LIMIT.NE.0) ENCODE (40,121,KAPTION) TP	LASER	992
C		LASER	993
C	CALCULATE EXTRAPOLATED VALUES OF THE POPULATION DENSITIES --	LASER	994
C		LASER	995
	SH = HOUT/H	LASER	996
	SJ = 1.	LASER	997
	JP1 = JSTART+1	LASER	998
	DO 25 I = 1,NP2	LASER	999
25	POP(I) = 0.	LASER	1000
	DO 65 J = 1,JP1	LASER	1001
	DO 66 I = 1,NP2	LASER	1002
66	POP(I) = POP(I) + DNYDTN(I,J)*SJ	LASER	1003
65	SJ = SJ*SH	LASER	1004
	NE = POP(2)	LASER	1005
C		LASER	1006
C	-----	LASER	1007
C	ELECTRON KINETICS ANALYSIS	LASER	1008
C	-----	LASER	1009
C		LASER	1010
	IF (.NOT.ELECT) GO TO 48	LASER	1011
C		LASER	1012
C	CALL *DNDT* TO DETERMINE S0 AND SB PRIOR TO CALLING THE BOLTZMANN	LASER	1013
C	ANALYSIS. THE ELECTRON SOURCE FUNCTION IS: SEXT(U) = S0*DELTA(U)	LASER	1014
C	+ SB*S(U).	LASER	1015
C		LASER	1016
	CALL DNDT (NEQ, TP, POP, NDOT)	LASER	1017
	IF (DU.EQ.0.) S0 = S0 + SB	LASER	1018
	IF (DU.EQ.0.) SB = 0.	LASER	1019
C		LASER	1020
	CHARGE = POP(NP1)	LASER	1021
	CURRENT = -POP(NP2)	LASER	1022
	RD = DIST/AREA/(EE*MU)	LASER	1023
	IF (NE.GT.0.) RD = RD/NE	LASER	1024
	VOLT = CURRENT*RD	LASER	1025
	ITER = ITHAX	LASER	1026
		LASER	1027

	IF (VOLT.EQ.0.) ITER = 0	LASER	1028
	EVCM = ABS(VOLT/DIST)	LASER	1029
	ESQ = EVCM*EVCM	LASER	1030
	DNEDT = NOUT(2)	LASER	1031
C		LASER	1032
	NOUT(NP,1,NP1) = VC = CHARGE/CAPAC/1000.	LASER	1033
	NOUT(NP,2,NP1) = KVOLT = VOLT/1000.	LASER	1034
	NOUT(NP,1,NP2) = VR = CURRENT*RESIST/1000.	LASER	1035
	NOUT(NP,2,NP2) = LDIDT = VC - VR - KVOLT	LASER	1036
C		LASER	1037
	CALL LEVELS (N1, N2, POP)	LASER	1038
C		LASER	1039
	IF (NP,EQ.1) GO TO 90	LASER	1040
	DP = NE*ELASTIC + EE*(UBAR*DNEDT - NE*NU*ESQ - UPLUS*SB)	LASER	1041
	DEPOSIT = IBEAM(NP)*DVDX	LASER	1042
	PDISCH = NE*EE*NU*ESQ	LASER	1043
	PCOLL = 0.	LASER	1044
	DO 524 J = 1,NK	LASER	1045
	IF (NEL(J).GE.0) GO TO 530	LASER	1046
	PWR = N1(J)*POWER(J)	LASER	1047
	GO TO 524	LASER	1048
530	PWR = EE*U(J)*(N1(J)*VSIG(1,J) - N2(J)*VSIG(2,J))	LASER	1049
524	PCOLL = PCOLL + PWR	LASER	1050
	DP = DP + NE*PCOLL	LASER	1051
	BEFORE = 100.*DP/(PDISCH + DEPOSIT)	LASER	1052
C		LASER	1053
	90 CALL SECOND (T0)	LASER	1054
C		LASER	1055
C	-----	LASER	1056
C		LASER	1057
	CALL BOLTZ (MGRID,1, MESH, NK, NAME, FI, NGAS, NMOL, TMOL, ITER,	LASER	1058
	1 THAX, EPS, KAPTION, TODAY, OUT, EVCM, NE, PROCESS, U, N1, N2,	LASER	1059
	2 NEL, S, SB, SO, EV, Q, QMOM, F, G, A, B, VSIG, POWER, PCOLL,	LASER	1060
	3 PDISCH, DEPOSIT, DEDT, ELASTIC, DNEDT, DLNEDT, IONIZE, ATTACH,	LASER	1061
	4 VD, MU, D, EK, AMPS, UBAR, TE, CONVRGE, PERCENT)	LASER	1062
C		LASER	1063
C	-----	LASER	1064
C		LASER	1065
	E(2) = UBAR	LASER	1066
	CALL SECOND (T1)	LASER	1067
	TBOLTZ = TBOLTZ + (T1 - T0)	LASER	1068
	NBOLTZ = NBOLTZ + 1	LASER	1069
C		LASER	1070
	IF (ITER.LE.0) PERCENT = 0.	LASER	1071
	REJECT = (.NOT.CONVRGE).OR.(PERCENT.GT.PCT)	LASER	1072
	REPEAT = REJECT.AND.EXPAND	LASER	1073
	IF (REPEAT) GO TO 95	LASER	1074
	IF (REJECT) GO TO 99	LASER	1075
C		LASER	1076
C	OBTAIN ELECTRON PARAMETERS, NORMALIZED WITHOUT POPULATION DENSITY:	LASER	1077
	IF (NE,NE.0.) ELASTIC = ELASTIC/NE	LASER	1078
	DO 523 J = 1,NK	LASER	1079
	IF (NEL(J).GE.0) GO TO 523	LASER	1080
	IF (N1(J).NE.0.) POWER(J) = POWER(J)/N1(J)	LASER	1081
523	IF (NE,NE.0.) POWER(J) = POWER(J)/NE	LASER	1082
C		LASER	1083
C	-----	LASER	1084



C		LASER	1085
C	THE RATES PRODUCED BY SUBROUTINE RATES HERE ARE USED FOR OUTPUT	LASER	1086
C	INFORMATION ONLY --	LASER	1087
C		LASER	1088
C	-----	LASER	1089
C	48 CALL DNDT (NEQ, TP, POP, NDOT)	LASER	1090
C	-----	LASER	1091
C		LASER	1092
C	(NOTE: DO NOT INTERCHANGE POP AND NO.)	LASER	1093
	RD = DIST/AREA/(EE*MU)	LASER	1094
	IF (NE.GT.0.) RD = RD/NE	LASER	1095
	COND(NP,1) = SIGMA = EE*MU*NE	LASER	1096
	COND(NP,2) = RD	LASER	1097
	JSUS = EE*NE*VD	LASER	1098
	JSUS(NP,1) = SIGMA*EVCM	LASER	1099
	JSUS(NP,2) = - NDOT(NP2)/AREA	LASER	1100
C		LASER	1101
C	E(1) = HNU	LASER	1102
C	CAVITY PHOTON DENSITY --	LASER	1103
	PHOTON = POP(1)	LASER	1104
	DNPHDT = NDOT(1)	LASER	1105
C		LASER	1106
C	EFFECTIVE PHOTON DENSITY IN MEDIUM --	LASER	1107
	NDOT(1) = (CAVITY/LENGTH)*NDOT(1)	LASER	1108
	POP(1) = (CAVITY/LENGTH)*POP(1)	LASER	1109
C		LASER	1110
	ETOT = DUDT = 0.	LASER	1111
	DO 86 I = 1, NTYPE	LASER	1112
	ITAU(I) = 1H	LASER	1113
	TAU = 0.	LASER	1114
	IF (NDOT(I).NE.0.) TAU = POP(I)/NDOT(I)	LASER	1115
	TAU = ABS(TAU)	LASER	1116
	TAU = TAU/UNIT	LASER	1117
	IF (TAU.NE.0.) ENCODE (10,FFMT,ITAU(I)) TAU	LASER	1118
	IF (TAU.GT.1.E 04) ENCODE (10,EFMT,ITAU(I)) TAU	LASER	1119
	NOUT(NP,1,I) = POP(I)	LASER	1120
	NOUT(NP,2,I) = NDOT(I)	LASER	1121
	ETOT = ETOT + E(I)*POP(I)	LASER	1122
A6	DUDT = DUDT + E(I)*NDOT(I)	LASER	1123
	ETOT = EE*ETOT	LASER	1124
	DUDT = EE*DUDT	LASER	1125
C		LASER	1126
	NDOT(1) = DNPHDT	LASER	1127
	POP(1) = PHOTON	LASER	1128
	RAD(NP,1) = RADIATE = 3.0E 10*HNU*PHOTON	LASER	1129
	RAD(NP,2) = 3.0E 10*HNU*DNPHDT	LASER	1130
	RAD(NP,3) = PBEAM = DVDX*IBEAM(NP)	LASER	1131
	PSTIM = GAMMA*RADIATE	LASER	1132
	HEAT = PDISCH + PBEAM - DUDT - PSTIM	LASER	1133
	P = PBEAM	LASER	1134
C		LASER	1135
	ALPHA(NP,1) = GNET	LASER	1136
	ALPHA(NP,2) = GAMMA	LASER	1137
	ALPHA(NP,3) = GAIN	LASER	1138
	ALPHA(NP,4) = ABSORB	LASER	1139
C		LASER	1140
	IF (.NOT.OUT(6)) GO TO 61	LASER	1141

C		LASER	1142
C		LASER	1143
C	-----	LASER	1144
C	OUTPUT OF POPULATION DENSITIES AND THEIR RATES OF CHANGE, AND	LASER	1145
C	MISCELLANEOUS ELECTRICAL AND OPTICAL PARAMETERS --	LASER	1146
C	-----	LASER	1147
C	ENCODE OPTICAL AND ELECTRICAL PARAMETERS --	LASER	1148
	L = 1	LASER	1149
	ENCODE (120,307,LINE(L)) IBEAM(NP), FACTOR, ENERGY	LASER	1150
	L = L+12	LASER	1151
	IF (ENERGY.LE.0.) L = L-4	LASER	1152
	DVDX = DVDX/1000.	LASER	1153
	PBEAM = PBEAM/1000.	LASER	1154
	ENCODE (80,325,LINE(L)) DVDX, PBEAM	LASER	1155
	L = L*8	LASER	1156
	ENCODE (80,308,LINE(L)) SB, S0	LASER	1157
	L = L*8	LASER	1158
	ENCODE (40,322,LINE(L))	LASER	1159
	L = L*4	LASER	1160
	IF (.NOT.ELECT) GO TO 53	LASER	1161
C		LASER	1162
	KVCM = KVOLT/DIST	LASER	1163
	PDISCH = PDISCH/1000.	LASER	1164
	P = P + PDISCH	LASER	1165
	IF (P.NE.0.) AFTER = PERCENT*(DEPOSIT + PDISCH)/P	LASER	1166
C		LASER	1167
	ENCODE (120,317,LINE(L)) AREA, DIST, COND(NP,1)	LASER	1168
	L = L+12	LASER	1169
	ENCODE (40,309,LINE(L)) RD	LASER	1170
	L = L*4	LASER	1171
	ENCODE (120,310,LINE(L)) CHARGE, CURRENT	LASER	1172
	L = L+12	LASER	1173
	ENCODE (120,311,LINE(L)) JSUS, KVCM, PDISCH	LASER	1174
	L = L+12	LASER	1175
	ENCODE (120,312,LINE(L)) VC, KVOLT	LASER	1176
	L = L+12	LASER	1177
	IF (RESIST.EQ.0.) GO TO 52	LASER	1178
	ENCODE (40,313,LINE(L)) VR	LASER	1179
	L = L*4	LASER	1180
52	IF (INDUCT.EQ.0.) GO TO 53	LASER	1181
	ENCODE (40,314,LINE(L)) LD10T	LASER	1182
	L = L*4	LASER	1183
53	IF (.NOT.STIM) GO TO 54	LASER	1184
	ENCODE (120,318,LINE(L)) REFLECT, PASS	LASER	1185
	L = L+12	LASER	1186
	IF (CAVITY.EQ.LENGTH) GO TO 504	LASER	1187
	ENCODE (80,324,LINE(L)) TCAVITY, CAVITY	LASER	1188
	L = L*8	LASER	1189
504	ENCODE (120,319,LINE(L)) LENGTH, OMEGA4P, GAMMA	LASER	1190
	L = L+12	LASER	1191
	ENCODE (120,321,LINE(L)) GNET, GAIN, ABSORB	LASER	1192
	L = L+12	LASER	1193
	ENCODE (40,322,LINE(L))	LASER	1194
	L = L*4	LASER	1195
	IF (P.EQ.0.) P = 1.E 99	LASER	1196
	PSTIM = PSTIM/1000.	LASER	1197
	EFF = 100.*PSTIM/P	LASER	1198

ENCODE (120,315,LINE(L)) RADIATE, PSTIM, EFF	LASER	1199
L = L*12	LASER	1200
IF (ABS(EFF).GT.100.) L = L-4	LASER	1201
C	LASER	1202
54 DEDT = DEDT/1000.	LASER	1203
DUOT = DUOT/1000.	LASER	1204
HEAT = HEAT/1000.	LASER	1205
ENCODE (120,316,LINE(L)) DEDT, DUOT, ETOT	LASER	1206
L = L*12	LASER	1207
ENCODE (40,327,LINE(L)) HEAT	LASER	1208
L = L*4	LASER	1209
IF (.NOT.ELECT) GO TO 529	LASER	1210
ENCODE (80,326,LINE(L)) BEFORE, AFTER	LASER	1211
L = L*8	LASER	1212
529 ENCODE (40,322,LINE(L))	LASER	1213
L = L*4	LASER	1214
ENCODE (120,320,LINE(L))	LASER	1215
LMAX = L*3	LASER	1216
C	LASER	1217
L = 0	LASER	1218
WRITE (6,110) KAPTION, UNIT	LASER	1219
DO 35 I = 1,NTYPE	LASER	1220
WRITE (6,400) I, GAS(I), E(I), POP(I), NDOT(I), ITAU(I),	LASER	1221
1 (LINE(L*K), K = 1,4)	LASER	1222
35 IF (L.LT.LMAX) L = L*4	LASER	1223
LC = NTYPE*6	LASER	1224
WRITE (6,401) (LINE(L*K), K = 1,4)	LASER	1225
L = L*4	LASER	1226
STEP = H/UNIT	LASER	1227
WRITE (6,107) STEP, (LINE(L*K), K = 1,4)	LASER	1228
L = L*4	LASER	1229
WRITE (6,114) JSTART, (LINE(L*K), K = 1,4)	LASER	1230
L = L*4	LASER	1231
WRITE (6,401) (LINE(L*K), K = 1,4)	LASER	1232
L = L*4	LASER	1233
WRITE (6,111) (LINE(L*K), K = 1,4)	LASER	1234
L = L*4	LASER	1235
51 IF (L.GE.LMAX) GO TO 77	LASER	1236
WRITE (6,401) (LINE(L*K), K = 1,4)	LASER	1237
LC = LC+1	LASER	1238
L = L*4	LASER	1239
GO TO 51	LASER	1240
77 NSKIP = 43-LC	LASER	1241
IF (NSKIP.LT.1) NSKIP = 1	LASER	1242
ENCODE (80,500,IMAGE) NSKIP	LASER	1243
WRITE (6,IMAGE) TODAY	LASER	1244
C	LASER	1245
C	LASER	1246
C	LASER	1247
C	LASER	1248
C	LASER	1249
61 IF (OUT(7).AND.NP.NE.1) CALL ANALYZE (INTYPE, KTYPE, RATEK, NTIME,	LASER	1250
1 RPCT, FLAG, PMAX, GAS, PER, KAPTION, LTAPE, MTAPE, NTAPE)	LASER	1251
C	LASER	1252
ARC = NE/NTOT.GT.RE	LASER	1253
END = END.OR.ARC	LASER	1254
IF (.NOT.END) GO TO 78	LASER	1255



	NF = NP-1	LASER	1256
	WRITE (6,225) NF, TP	LASER	1257
	IF (NBOLTZ.NE.0) WRITE (6,127) TA, NBOLTZ, TBOLTZ	LASER	1258
	IF (ARC) WRITE (6,224) NE, RE	LASER	1259
	GO TO 97	LASER	1260
C	-----	LASER	1261
C	CONTINUE TO INTEGRATE THE EQUATIONS FROM A DEAD START FROM THIS	LASER	1262
C	CYCLE POINT --	LASER	1263
C	-----	LASER	1264
C		LASER	1265
	78 T = TP )	LASER	1266
	DO 49 I = 1,NEQ	LASER	1267
	49 DNYDTN(I,1) = POP(I)	LASER	1268
	JSTART = NFLAG = 0	LASER	1269
	DELTA = ETA	LASER	1270
C		LASER	1271
C	-----	LASER	1272
	18 CALL GEAR (NEQ, T, DNYDTN, SCRATCH, H, HMIN, HMAX, DELTA, METHOD,	LASER	1273
	1 YMAX, ERR, KFLAG, JSTART, MAXDER, NMAXP2, PHI)	LASER	1274
C	-----	LASER	1275
C		LASER	1276
	TT = T - TP	LASER	1277
	IF (KFLAG.EQ.1) GO TO 30	LASER	1278
	JSTART = 0	LASER	1279
	NFLAG = NFLAG+1	LASER	1280
	IF (NFLAG.EQ.1) GO TO 18	LASER	1281
	H = H/100.	LASER	1282
	IF (NFLAG.LE.5) GO TO 18	LASER	1283
	DELTA = 2.*DELTA	LASER	1284
	IF (NFLAG.LE.10) GO TO 18	LASER	1285
C		LASER	1286
	97 IF (NP.LT.15) GO TO 99	LASER	1287
C		LASER	1288
C	-----	LASER	1289
C	OUTPUT GENERATION	LASER	1290
C	-----	LASER	1291
C		LASER	1292
C	SUMMARY OF UNIMPORTANT REACTIONS --	LASER	1293
C		LASER	1294
	REWIND MTAPE	LASER	1295
	KOUNT = 25	LASER	1296
	LC = 0	LASER	1297
	DO 507 K = 1,KTYPE	LASER	1298
	READ (MTAPE) (NTIME(I), I = 1,NTYPE), RATE, RNAME	LASER	1299
	IF (.NOT.FLAG(K)) GO TO 507	LASER	1300
	IF (LC.EQ.0) WRITE (6,402) PER	LASER	1301
	LC = LC+1	LASER	1302
	IF (LC.EQ.KOUNT) LC = 0	LASER	1303
	WRITE (6,403) K, RATE, (RNAME(L), L = 1,4)	LASER	1304
	507 CONTINUE	LASER	1305
C		LASER	1306
C	SUMMARY OF IMPORTANT REACTIONS --	LASER	1307
C		LASER	1308
	REWIND MTAPE	LASER	1309
	LC = 0	LASER	1310
	DO 501 K = 1,KTYPE	LASER	1311
		LASER	1312

READ (MTAPE) (NTIME(I), I = 1, NTYPE), RATE, RNAME	LASER	1313
IF (FLAG(K)) GO TO 501	LASER	1314
IF (LC.EQ.0) WRITE (6,404) PER	LASER	1315
LC = LC+1	LASER	1316
IF (LC.EQ.KOUNT) LC = 0	LASER	1317
WRITE (6,403) K, RATE, (RNAME(L), L = 1,4)	LASER	1318
501 CONTINUE	LASER	1319
C	LASER	1320
WRITE (6,216)	LASER	1321
CALL PLOT (M1, NP, 1, IBEAM, 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1322
1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, 1, 0)	LASER	1323
WRITE (6,306) UNIT, TODAY	LASER	1324
C	LASER	1325
IF (.NOT.STIM) GO TO 40	LASER	1326
KAPTION(1) = 10HINTENSITY	LASER	1327
KAPTION(2) = 5HDI/OT	LASER	1328
WRITE (6,218)	LASER	1329
CALL PLOT (M1, NP, 1, RAD, 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1330
1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, 2, 0)	LASER	1331
WRITE (6,306) UNIT, TODAY	LASER	1332
C	LASER	1333
40 KAPTION(1) = 10HNET GAIN	LASER	1334
KAPTION(2) = 10HRESHHOLD	LASER	1335
NPLOT = 2	LASER	1336
IF (.NOT.STIM) NPLOT = 1	LASER	1337
WRITE (6,113)	LASER	1338
CALL PLOT (M1, NP, 1, ALPHA, 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1339
1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)	LASER	1340
WRITE (6,306) UNIT, TODAY	LASER	1341
C	LASER	1342
KAPTION(1) = 10HLASER GAIN	LASER	1343
KAPTION(2) = 10HABSORPTION	LASER	1344
WRITE (6,126)	LASER	1345
CALL PLOT (M1, NP, 1, ALPHA(1,3), 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1346
1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, 2, 0)	LASER	1347
WRITE (6,306) UNIT, TODAY	LASER	1348
C	LASER	1349
NI = 0	LASER	1350
IF (.NOT.STIM) GO TO 502	LASER	1351
NI = NI+1	LASER	1352
KAPTION(1) = 10HOPTICAL	LASER	1353
DO 82 I = 1, NP	LASER	1354
ALPHA(I,3) = ALPHA(I,4) = 0.	LASER	1355
RAD(I,1) = ALPHA(I,1)*RAD(I,1)	LASER	1356
82 ALPHA(I,1) = RAD(I,1)/1000.	LASER	1357
502 IF (.NOT.ELECT) GO TO 511	LASER	1358
NI = NI+1	LASER	1359
KAPTION(NI) = 10HELECTRICAL	LASER	1360
DO 88 I = 1, NP	LASER	1361
KVCM = NOUT(I,2,NP1)/DIST	LASER	1362
AMPS = ISUS(I,1)	LASER	1363
ALPHA(I,4) = KVCM*AMPS	LASER	1364
88 RAD(I,NI) = 1000.*ALPHA(I,4)	LASER	1365
511 IF (JBEAM.EQ.0.) GO TO 96	LASER	1366
NI = NI+1	LASER	1367
KAPTION(NI) = 10ME-BEAM	LASER	1368
DO 512 I = 1, NP	LASER	1369

	RAD(I,NI) = RAD(I,3)	LASER	1370
512	ALPHA(I,3) = RAD(I,3)/1000.	LASER	1371
96	IF (NI.EQ.0) GO TO 508	LASER	1372
	NPLOT = NI	LASER	1373
	WRITE (6,122)	LASER	1374
	CALL PLOT (M1, NP, 1, RAD, 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1375
1	.TRUE., .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)	LASER	1376
	WRITE (6,306) UNIT, TODAY	LASER	1377
C		LASER	1378
	IF (.NOT.STIM) GO TO 509	LASER	1379
C	COMPUTE POWER EFFICIENCY --	LASER	1380
	DO 510 I = 1,NP	LASER	1381
	P = RAD(I,3)	LASER	1382
	IF (ELECT) P = P + RAD(I,2)	LASER	1383
	IF (P.EQ.0.) P = 1.E 99	LASER	1384
	ALPHA(I,2) = 100.*RAD(I,1)/P	LASER	1385
510	IF (ABS(ALPHA(I,2)).GT.100.) ALPHA(I,2) = 0.	LASER	1386
C		LASER	1387
C	-----	LASER	1388
C	INTEGRATE POWER DENSITIES --	LASER	1389
C	-----	LASER	1390
		LASER	1391
509	DO 506 I = 1,3	LASER	1392
	B1 = RAD(I,1)	LASER	1393
	RAD(I,1) = 0.	LASER	1394
	DO 506 K = 2,NP	LASER	1395
	K1 = K-1	LASER	1396
	B2 = B1	LASER	1397
	B1 = RAD(K,1)	LASER	1398
506	RAD(K,1) = RAD(K1,1) + 0.5E 03*TOU*(B1 + B2)	LASER	1399
C		LASER	1400
	WRITE (6,125)	LASER	1401
	CALL PLOT (M1, NP, 1, RAD, 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1402
1	.TRUE., .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)	LASER	1403
	WRITE (6,306) UNIT, TODAY	LASER	1404
C		LASER	1405
C	-----	LASER	1406
C	TABULAR SUMMARY OF MISCELLANEOUS ELECTRICAL AND OPTICAL PARAMETERS	LASER	1407
C	-----	LASER	1408
		LASER	1409
	12 = 1	LASER	1410
522	11 = 12	LASER	1411
	12 = 12*50	LASER	1412
	IF (12.GT.NP) 12 = NP	LASER	1413
	WRITE (6,223) UNIT	LASER	1414
	WRITE (6,226) (TIME(1), IBEAM(1), ALPHA(I,3), NOUT(1,2,NP1),	LASER	1415
1	ISUS(I,1), ALPHA(I,4), ALPHA(I,1), ALPHA(I,2), I = 11,12)	LASER	1416
	IF (12.LT.NP) GO TO 522	LASER	1417
C		LASER	1418
	IF (.NOT.STIM) GO TO 508	LASER	1419
	DO 513 I = 1,NP	LASER	1420
	ALPHA(I,3) = RAD(I,1)	LASER	1421
	P = RAD(I,3)	LASER	1422
	IF (ELECT) P = P + RAD(I,2)	LASER	1423
	IF (P.EQ.0.) P = 1.E 99	LASER	1424
513	ALPHA(I,4) = 100.*RAD(I,1)/P	LASER	1425
C		LASER	1426



C	-----	LASER	1427
C	PLOT OPTICAL POWER DENSITY AND EFFICIENCY --	LASER	1428
C	-----	LASER	1429
C		LASER	1430
	KAPTION(1) = 10HPOWER/VOL	LASER	1431
	KAPTION(2) = 10HEFFICIENCY	LASER	1432
	WRITE (6,221)	LASER	1433
	CALL PLOT (M1, NP, 1, ALPHA, 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1434
	1 .TRUE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)	LASER	1435
	WRITE (6,306) UNIT, TODAY	LASER	1436
C		LASER	1437
C	-----	LASER	1438
C	PLOT OPTICAL ENERGY DENSITY AND EFFICIENCY --	LASER	1439
C	-----	LASER	1440
C		LASER	1441
	KAPTION(1) = 10HENERGY/VOL	LASER	1442
	KAPTION(2) = 10HEFFICIENCY	LASER	1443
	WRITE (6,222)	LASER	1444
	CALL PLOT (M1, NP, 1, ALPHA(1,3), 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1445
	1 .TRUE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)	LASER	1446
	WRITE (6,306) UNIT, TODAY	LASER	1447
C		LASER	1448
	508 IF (.NOT.ELECT) GO TO 85	LASER	1449
	KAPTION(1) = 4HISUS	LASER	1450
	KAPTION(2) = 8HDISUS/DT	LASER	1451
	WRITE (6,214)	LASER	1452
	CALL PLOT (M1, NP, 1, ISUS, 0., 0., TIME, 0., DTIME, .FALSE.,	LASER	1453
	1 .TRUE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)	LASER	1454
	WRITE (6,306) UNIT, TODAY	LASER	1455
C		LASER	1456
	KAPTION(1) = 6HSIGMA	LASER	1457
	KAPTION(2) = 10HRD (OHM)	LASER	1458
	Y0(1) = Y0(2) = DY(1) = 0.	LASER	1459
	DY(2) = 0.5	LASER	1460
	WRITE (6,215)	LASER	1461
	CALL PLOT (M1, NP, 1, COND, Y0, DY, TIME, 0., DTIME, .FALSE.,	LASER	1462
	1 .FALSE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)	LASER	1463
	WRITE (6,306) UNIT, TODAY	LASER	1464
C		LASER	1465
	NPLOT = 4	LASER	1466
	KAPTION(1) = 10HCAPACITOR	LASER	1467
	KAPTION(2) = 10HDISCHARGE	LASER	1468
	KAPTION(3) = 10HRESISTANCE	LASER	1469
	KAPTION(4) = 10HINDUCTANCE	LASER	1470
C		LASER	1471
	IF (INDUCT.EQ.0.) NPLOT = 3	LASER	1472
	VMIN = 0.	LASER	1473
	IF (NOUT(NP,1,NP1).LT.VMIN) VMIN = NOUT(NP,1,NP1)	LASER	1474
	IF (NOUT(NP,2,NP1).LT.VMIN) VMIN = NOUT(NP,2,NP1)	LASER	1475
	IF (NOUT(NP,1,NP2).LT.VMIN) VMIN = NOUT(NP,1,NP2)	LASER	1476
	IF (NOUT(NP,2,NP2).LT.VMIN) VMIN = NOUT(NP,2,NP2)	LASER	1477
	CALL AXIS (1, .TRUE., VMAX, VMIN, Y0, DY, YDC)	LASER	1478
	Y0(2) = DY(2) = 0.	LASER	1479
	TEST = YDC.NE.0.	LASER	1480
C		LASER	1481
	WRITE (6,217)	LASER	1482
	CALL PLOT (M1, NP, 1, NOUT(1,1,NP1), Y0, DY, TIME, 0., DTIME,	LASER	1483

1	.FALSE., TEST, .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)	LASER	1484
	WRITE (6,306) UNIT, TODAY	LASER	1485
C		LASER	1486
85	DO 50 I = 2, NTYPE	LASER	1487
	IF (.NOT. PLOTS(I)) GO TO 50	LASER	1488
	ENCODE (20,211, KAPTION) I, I	LASER	1489
	WRITE (6,227) GAS(I)	LASER	1490
	CALL PLOT (M1, NP, 1, NOUT(1,1,I), 0., 0., TIME, 0., DTIME,	LASER	1491
	1 .FALSE., .TRUE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)	LASER	1492
	WRITE (6,306) UNIT, TODAY	LASER	1493
50	CONTINUE	LASER	1494
C		LASER	1495
C	----- FORMAT STATEMENTS -----	LASER	1496
C		LASER	1497
100	FORMAT (80A1)	LASER	1498
C		LASER	1499
101	FORMAT (8A10)	LASER	1500
C		LASER	1501
102	FORMAT (/1X,134(1H-)/)	LASER	1502
C		LASER	1503
103	FORMAT (18X,*SUMMARY OF UPDATED RATES FOR INPUT REACTION SCHEME OF	LASER	1504
	1 SYNTHETIC KINETICS CODE GENERATED ON* A10/25X* DR. WILLIAM B. LACIN	LASER	1505
	2A, NORTHROP RESEARCH AND TECHNOLOGY CENTER, PALOS VERDES, CALIFORN	LASER	1506
	3IA*)	LASER	1507
C		LASER	1508
104	FORMAT (/1X,A2,I4,3X,*A10,A5,2X,A10,3X,A10,5X,5A10)	LASER	1509
C		LASER	1510
105	FORMAT (/6X*I*6X*REACTION(I)*36X*KF(I)*8X*KR(I)*20X*REFERENCES OR	LASER	1511
	2COMMENTS*/1X,134(1H-))	LASER	1512
C		LASER	1513
106	FORMAT (*MOMENTUM TRANSFER FOR *,A10)	LASER	1514
C		LASER	1515
107	FORMAT (43X*INTEGRATION STEP SIZE =*F10.3,6X,*A10)	LASER	1516
C		LASER	1517
108	FORMAT ((30X,*A10))	LASER	1518
C		LASER	1519
109	FORMAT (*(1H1*I2*(/))*)	LASER	1520
C		LASER	1521
110	FORMAT (1H1/55X,*A10///24X*POPULATION DENSITIES AND RATES OF CHANG	LASER	1522
	1E*20X*ELECTRICAL AND OPTICAL PARAMETERS*//13X*I*5X*SPECIES*6X*E(I)	LASER	1523
	2*8X*N(I)*7X*DN(I)/DT*7X*TAU(I)*//20X*NAME*8X*(EV)*7X*(CM-3)*5X*(CM-	LASER	1524
	33/SEC)*1PE10.1* SEC*7X*PARAMETER*6X*VALUE*5X*UNITS*/9X,70(1H-),3X,	LASER	1525
	440(1H-)/)	LASER	1526
C		LASER	1527
111	FORMAT (9X,70(1H-),3X,*A10)	LASER	1528
C		LASER	1529
112	FORMAT (10X,3E10.3)	LASER	1530
C		LASER	1531
113	FORMAT (1H1/53X*INSTANTANEOUS NET GAIN COEFFICIENT (CM-1)*//)	LASER	1532
C		LASER	1533
114	FORMAT (44X*ORDER OF INTEGRATION =*I10,6X,*A10)	LASER	1534
C		LASER	1535
115	FORMAT (10X*NO E- CROSS SECTION DATA WAS FOUND.*)	LASER	1536
C		LASER	1537
116	FORMAT (10X*SIGMA = 0 IN THE RANGE (0,*F4.1*) EV.*)	LASER	1538
C		LASER	1539
117	FORMAT (10X*ERRORS OCCURRED IN CROSS SECTION DATA.*)	LASER	1540



C	118 FORMAT (10X*CROSS SECTION DOES NOT SPAN (0,*F4.1*) EV.*)	LASER	1541
C	119 FORMAT (10X*THE MASS ENTERED FOR THIS SPECIES IS ≤ 0*)	LASER	1542
C	120 FORMAT (A10* * E * E * *A10)	LASER	1543
C	121 FORMAT (8X*TIME T = *.1PE10.3* SEC*)	LASER	1544
C	122 FORMAT (1H1,56X*POWER DENSITY (WATT/CM3)*//)	LASER	1545
C	123 FORMAT (45X* THE ORIGINAL RATE CONSTANT(S) HAVE BEEN MODIFIED#)	LASER	1546
C	124 FORMAT (44X** ILLEGAL ATTEMPT TO MODIFY RATE CONSTANT(S) WAS REJ 1ECTED#)	LASER	1547
C	125 FORMAT (1H1,56X*ENERGY DENSITY (JOULE/LITER)*//)	LASER	1548
C	126 FORMAT (1H1,42X*LASER GAIN AND MEDIUM ABSORPTION COEFFICIENTS (CM- 11)*//)	LASER	1549
C	127 FORMAT (40X*THE PROCESSING OF ELECTRON CROSS SECTIONS REQUIRED*, 1F5.1* CP SEC.*/40X*AND*13* BOLTZMANN ELECTRON CALCULATIONS CONSUME 2D*F6.1* CP SEC.*)	LASER	1550
C	128 FORMAT (50X*NO FURTHER WARNING DIAGNOSTICS WILL BE ISSUED.*)	LASER	1551
C	129 FORMAT (12*) *A10.5X)	LASER	1552
C	130 FORMAT (1H1,24X,*NORMALIZED EXTERNAL SOURCE FUNCTION FOR CREATION 1OF ELECTRONS IN THE ENERGY RANGE (U, U + DU)*/43X*AVERAGE ENERGY 0 2F CREATED ELECTRONS = < U* > = *F6.2* EV*/)	LASER	1553
C	131 FORMAT (1/62X*ELECTRON ENERGY U (EV)*)	LASER	1554
C	201 FORMAT (//15X,A10*ERRORS WERE DETECTED IN PROCESSING THE INPUT REA 1TION SCHEME. MODIFICATIONS OF THE REACTION*/25X*SCHEME, CORRECTIO 2NS IN REACTION SYNTAX, CHANGES IN DIMENSION STORAGE, OR ADDITIONS 3TO THE E-*/25X*CROSS SECTION FILE MAY BE REQUIRED TO REMOVE ALL OF 4 THE ERROR CONDITIONS.*)	LASER	1555
C	202 FORMAT (//15X,A10*THE FOLLOWING SPECIES WERE INITIALIZED BY INPUT 1DATA, BUT DID NOT OCCUR IN THE KINETIC SYSTEM.*/25X*IF THE PROGRAM 2 IS EXECUTED, THEY WILL BE IGNORED, BUT INCLUDED IN THE TOTAL PRES 3SURE (AS BUFFER*/25X*GASES IN 3-BODY COLLISIONS) AND FOR MOMENTUM 4TRANSFER IN E- KINETICS ANALYSIS:*/)	LASER	1556
C	203 FORMAT (//15X,A10*INITIAL POPULATION DENSITIES FOR THE FOLLOWING S 1PECIES WERE NOT DEFINED. IF THE PROGRAM IS EXE-*/25X*CUTED, NO(1) 2 = 0 WILL BE ASSUMED:*/)	LASER	1557
C	204 FORMAT (//15X,A10*ENERGIES FOR THE FOLLOWING SPECIES WERE NOT DEFI 1NED. IF PROGRAM IS EXECUTED, E(1) = 0 WILL RE*/25X*ASSUMED:*/)	LASER	1558
C	205 FORMAT (//15X,A10*ERRORS OCCURRED FOR THE INPUT CROSS SECTIONS FOR 1 THE FOLLOWING ELECTRON COLLISION PROCESSES.*/25X*IF THE PROGRAM I 2S EXECUTED, A ZERO CROSS SECTION OVER ALL ENERGY WILL BE ASSUMED:*	LASER	1559

	3/1	LASER	1598
C	206 FORMAT ((40X,5A10))	LASER	1599
		LASER	1600
C	207 FORMAT (//15X,A10*AN ATTEMPT WAS MADE TO ENTER. BY \$RATES ... \$ IN	LASER	1601
	1PUT. THE FOLLOWING RATES, WHICH ARE NOT ACCESSIBLE BY INPUT	LASER	1602
	2 FOR THE SYNTHETIC PROGRAM THAT WAS GENERATED. IF THE PROGRAM EXE	LASER	1603
	3CUTES. THE ATTEMPTED MODIFICATIONS WILL BE IGNORED, AND THE	LASER	1604
	4 ORIGINAL RATES USED:*)	LASER	1605
		LASER	1606
C	208 FORMAT (//15X,A10*SYNTHESIZED SUBROUTINES AND DATA FILE WERE GENER	LASER	1607
	ATED WITH DIMENSION DECLARATORS NMAX =*I4*,*/25X*KMAX =*I4*, AND N	LASER	1608
	2KMAX =*I4*. THESE MUST AGREE WITH THE CORRESPONDING DIMENSION DEC	LASER	1609
	3LARATORS */25X*IN THE MAIN PROGRAM (LASER). EXECUTION MAY BE POSS	LASER	1610
	4IBLE IF PRESENT STORAGE EXCEEDS ORIGINAL. BUT*/25X*CAUTION IS ADVI	LASER	1611
	5VISED TO INSURE THAT LABELED COMMON BLOCKS AGREE WITH THOSE IN THE	LASER	1612
	6 SYNTHETIC*/25X*SUBROUTINES ASSOCIATED WITH THE DATA FILE GENERATE	LASER	1613
	7D ON TAPE4.*)	LASER	1614
		LASER	1615
C	209 FORMAT (//15X,A10*TAPE4 DATA FILE VECTORS EXCEEDED DIMENSION STORA	LASER	1616
	1GE IN MAIN PROGRAM, AND WERE TRUNCATED DURING*/25X*READ. NKMAX =*	LASER	1617
	2I4*, KMAX =*I4*, AND NMAX =*I4*. TAPE4 CONTAINS NK =*I4*, KTYPE =	LASER	1618
	3*I4*, AND*/25X*NTYPE =*I4*. CAUTION IS ADVISED.*)	LASER	1619
		LASER	1620
C	210 FORMAT (//15X,A10*MOMENTUM TRANSFER COLLISION FREQUENCY IS ZERO AT	LASER	1621
	1 SOME POINT. E- ANALYSIS CONTAINS 1/QM TERMS.*)	LASER	1622
		LASER	1623
C	211 FORMAT (2HN(,I2,1H).5X,3HDN(,I2,4H)/DT,1X)	LASER	1624
		LASER	1625
C	212 FORMAT (1H1/55X*SUMMARY OF POSSIBLE ERROR CONDITIONS*////15X*SEVER	LASER	1626
	1ITY*45X*DESCRIPTION */15X,105(1H-))	LASER	1627
		LASER	1628
C	214 FORMAT (1H1,25X*DISCHARGE CURRENT DENSITY JSUS (AMP/CM2), AND ITS	LASER	1629
	1RATE OF CHANGE D/DT(JSUS) (A/CM2/S)*)	LASER	1630
		LASER	1631
C	215 FORMAT (1H1,20X*PLASMA CONDUCTIVITY SIGMA = E*NE*MU (1/CM/OHM), AND	LASER	1632
	1 DISCHARGE IMPEDANCE RD = D/AREA/SIGMA (OHM)*)	LASER	1633
		LASER	1634
C	216 FORMAT (1H1/40X*E-BEAM CURRENT DENSITY (AMP/CM2) AS A FUNCTION OF	LASER	1635
	1TIME*)	LASER	1636
		LASER	1637
C	217 FORMAT (1H1/56X*CIRCUIT VOLTAGES (KVOLT)*)	LASER	1638
		LASER	1639
C	218 FORMAT (1H1/27X*INTRACAVITY RADIATION INTENSITY I (W/CM2) AND ITS	LASER	1640
	1RATE OF CHANGE DI/DT (W/CM2/SEC)*)	LASER	1641
		LASER	1642
C	219 FORMAT (1JX*(VSIG(K,I) ARE FORWARD (K=1) OR REVERSE (K=2) RATES F	LASER	1643
	1OR THE ITH INELASTIC PROCESS IN THE E- KINETICS ANALYSIS*)	LASER	1644
		LASER	1645
C	220 FORMAT (1H1,20(//),40X*ELECTRON ENERGY RANGE EMAX =*.0PF6.2* EXCEED	LASER	1646
	1S MAXIMUM VALUE =*.0PF6.2)	LASER	1647
		LASER	1648
C	221 FORMAT (1H1,40X*OPTICALLY EXTRACTED POWER DENSITY (KW/CM3) AND EFF	LASER	1649
	1ICIENCY (%)*)	LASER	1650
		LASER	1651
C	222 FORMAT (1H1,35X*OPTICALLY EXTRACTED ENERGY DENSITY (JOULE/LITER) A	LASER	1652
	1ND EFFICIENCY (%)*)	LASER	1653
		LASER	1654



C	223	FORMAT (1H1,45X*SUMMARY OF ELECTRICAL AND OPTICAL PARAMETERS*//	LASER	1655
		123X*TIME*7X*JBEAM*7X*PBEAM*6X*VOLTAGE*7X*JSUS*7X*DISCH*7X*OUTPUT*	LASER	1656
		26X*OPT EFF*/18X*(*,1PE8.2*,* SEC)* 1X*(A/CM2)*4X*(KW/CM3)*7X*(KV)*,	LASER	1657
		36X*(A/CM2)*4X*(KW/CM3)*5X*(KW/CM3)*7X*(%)*18X.99(1H-1)/)	LASER	1658
			LASER	1659
C	224	FORMAT (40X*ELECTRON DENSITY NE =*1PE10.3* EXCEEDS LIMIT NE/NMOL ≤	LASER	1660
		1 *1PE10.3)	LASER	1661
			LASER	1662
C	225	FORMAT (1/40X*CALCULATION IS TERMINATED AT CYCLE NP =*I3*, T = *1PE	LASER	1663
		110.3*,* SEC.*)	LASER	1664
			LASER	1665
C	226	FORMAT (18X,F9.2,5F12.2,F13.2,F11.2)	LASER	1666
			LASER	1667
C	227	FORMAT (1H1/32X,*POPULATION DENSITY N (CM-3) AND RATE OF CHANGE DN	LASER	1668
		1/DT (CM-3/S) FOR *,A10/)	LASER	1669
			LASER	1670
C	300	FORMAT (///25X*PROGRAM IS TERMINATED FOR ERRORS SPECIFIED FATAL.*)	LASER	1671
			LASER	1672
C	301	FORMAT (1PE10.4)	LASER	1673
			LASER	1674
C	302	FORMAT (///25X.86(1H*)///)	LASER	1675
			LASER	1676
C	303	FORMAT (28X.8A10)	LASER	1677
			LASER	1678
C	304	FORMAT (1H1,5(/))	LASER	1679
			LASER	1680
C	305	FORMAT (*K*,A1,*(*,I3*,*))	LASER	1681
			LASER	1682
C	306	FORMAT (1/62X*TIME (*1PE9.3* SEC)*100X*DR. WILLIAM B. LACINA,*A11/	LASER	1683
		1100X*NORTHROP RESEARCH AND TECHNOLOGY*)	LASER	1684
			LASER	1685
C	307	FORMAT (13X*JBEAM*5X* = *0PF10.2* AMP/CM2*,8X*DEPOSITION = *F10.2,	LASER	1686
		117X*ENERGY*4X* = *F10.0* KEV*,9X)	LASER	1687
			LASER	1688
C	308	FORMAT (13X*S(U > 0) = *1PE10.3* CM-3/SEC*7X*S(U = 0) = *	LASER	1689
		1 1PE10.3* CM-3/SEC*4X)	LASER	1690
			LASER	1691
C	309	FORMAT (13X*R(DISCH)*2X* = *1PE10.3* OHM*)	LASER	1692
			LASER	1693
C	310	FORMAT (43X*Q*9X* = *,1PE10.3* COULOMB*8X*I(DISCH)*2X* = *,1PE10.	LASER	1694
		13* AMP*9X)	LASER	1695
			LASER	1696
C	311	FORMAT (13X*J(DISCH)*2X* = *1PE10.3* AMP/CM2*8X*ESUS*6X* = *0PF10.	LASER	1697
		13* KVOLT/CM*7X*ESUS*JSUS = *0PF10.2* KW/CM3*)	LASER	1698
			LASER	1699
C	312	FORMAT (43X*Q/C*7X* = *0PF10.3* KVOLT*10X*V(DISCH)*2X* = *0PF10.3	LASER	1700
		1* KVOLT*)	LASER	1701
			LASER	1702
C	313	FORMAT (13X*V(RESIST) = *0PF10.3* KVOLT*)	LASER	1703
			LASER	1704
C	314	FORMAT (13X*L*DI/DT*3X* = *0PF10.3* KVOLT*)	LASER	1705
			LASER	1706
C	315	FORMAT (13X*INTENSITY = *1PE10.3* WATT/CM2*,7X*OPTICAL*3X* = *,	LASER	1707
		10PF10.2* KW/CM3*9X*EFFICIENCY = *,0PF10.2* %)	LASER	1708
			LASER	1709
C	316	FORMAT (13X*E<U>DNE/DT = *1PE10.3* KW/CM3*9X*DE(TOT)/DT = *E10.3,	LASER	1710
			LASER	1711



	1* KW/CM3*9X*E(TOT)*4X* = *E10.3* J/CM3*7X)	LASER	1712
C		LASER	1713
	317 FORMAT (3X*AREA*6X* = *0PF10.2* CM2*12X*DIST*6X* = *0PF10.2* CM*	LASER	1714
	113X*CONDUCT*3X* = *1PE10.3* /OHM/CM*)	LASER	1715
C		LASER	1716
	318 FORMAT (43X*REFLECT*3X* = *0PF10.2* %*14X*LOSS*6X* = *0PF10.2* %	LASER	1717
	1/PASS*)	LASER	1718
C		LASER	1719
	319 FORMAT (3X*LENGTH*4X* = *0PF10.2* CM*13X*OMEGA/4/PI = *1PE10.3,	LASER	1720
	117X*THRESH*4X* = *1PE10.3* CM-1*)	LASER	1721
C		LASER	1722
	320 FORMAT (40(1H-),80X)	LASER	1723
C		LASER	1724
	321 FORMAT (3X*NET GAIN = *1PE10.3,* CM-1*11X*LASER GAIN = *1PE10.3	LASER	1725
	1* CM-1*11X*ABSORPTION = *.1PE10.3* CM-1*)	LASER	1726
C		LASER	1727
	322 FORMAT (10X)	LASER	1728
C		LASER	1729
	323 FORMAT (***** THE ORIGINAL RATE HAS BEEN MODIFIED *****)	LASER	1730
C		LASER	1731
	324 FORMAT (3X*T(CAVITY) = *F10.1* NS*13X*L(CAVITY) = *F10.2* CM*)	LASER	1732
C		LASER	1733
	325 FORMAT (3X*DVOX*6X* = *F10.2* KV/CM*10X*P(BEAM)*3X* = *F10.2,	LASER	1734
	1* KW/CM3*6X)	LASER	1735
C		LASER	1736
	326 FORMAT (3X*DP(BEFORE) = *F10.2* %*14X*DP(AFTER) = *F11.2* %*11X)	LASER	1737
C		LASER	1738
	327 FORMAT (3X*HEAT * SP = *1PE10.3* KW/CM3*6X)	LASER	1739
C		LASER	1740
	400 FORMAT (9X,I5,5X,A10,0PF7.2,1P2E14.3,2X,A10,6X,4A10)	LASER	1741
C		LASER	1742
	401 FORMAT (82X,4A10)	LASER	1743
C		LASER	1744
	402 FORMAT (1H1,36X*THE FOLLOWING REACTIONS CONTRIBUTE LESS THAN *F3.0	LASER	1745
	1* %//29X*K*8X*KF(K)*10X*KR(K)*15X*REACTION(K)*27X,75(1H-)/)	LASER	1746
C		LASER	1747
	403 FORMAT (25X,I5,5X,A10,5X,A10,5X,4A10/)	LASER	1748
C		LASER	1749
	404 FORMAT (1H1,40X*THE FOLLOWING REACTIONS WERE IMPORTANT (> *F3.0,	LASER	1750
	1* %//29X*K*8X*KF(K)*10X*KR(K)*15X*REACTION(K)*27X,75(1H-)/)	LASER	1751
C		LASER	1752
	500 FORMAT (1H(,12,*(//)90X*DR. WILLIAM B. LACINA,*A11/90X*NORTHROP RES	LASER	1753
	1EARCH AND TECHNOLOGY*)#)	LASER	1754
C		LASER	1755
	-----	LASER	1756
C		LASER	1757
	99 CALL EXIT	LASER	1758
	END	LASER	1759

	SUBROUTINE SYNTH (LTAPE, MTAPE, NTAPE, NSCRTCH, NDATA, NSIZE,	SYNTH	2
	1 MAXGAS, GAS, KMAX, NKMAX, LEV1, LEV2, DATE)	SYNTH	3
C	.....	SYNTH	4
C		SYNTH	5
C		SYNTH	6
C	THIS SUBROUTINE WILL EDIT THE INPUT FILE OF KINETIC REACTIONS AND	SYNTH	7
C	AUTOMATICALLY GENERATE SUBROUTINES REQUIRED FOR THE MOLECULAR KI-	SYNTH	8
C	NETICS ANALYSIS AND ITS LINKAGE TO COUPLED ELECTRON KINETICS CAL-	SYNTH	9
C	CULATIONS. SYMBOLIC REACTIONS ARE TRANSLATED INTO COMPUTER CODED	SYNTH	10
C	EQUATIONS. SUBROUTINES SYNTHESIZED ARE: (1) RATES OF CHANGE $DN_i/DT$	SYNTH	11
C	OF POPULATION DENSITIES, (2) THE JACOBIAN $D(DN_i/DT)/D(N_j)$ , AND (3)	SYNTH	12
C	DEFINITION OF THE CORRESPONDENCE OF MOLECULAR STATES WHICH OCCUR	SYNTH	13
C	IN THE ELECTRON-MOLECULE SCATTERING PROCESSES FOR THE E- KINETICS.	SYNTH	14
C	INPUT CONSISTS OF AN ARBITRARILY LONG SEQUENCE OF CARD PAIRS OF	SYNTH	15
C	THE FORM	SYNTH	16
C		SYNTH	17
C	1) A1 * A2 * A3 * ... * B1 * B2 * B3 * ...	SYNTH	18
C	2) KF, KR, KOMMENT (2E10.3,5X,5A10)	SYNTH	19
C		SYNTH	20
C	WHERE A1, A2, ..., B1, B2, ... ARE THE PHYSICAL NAMES OF THE RE-	SYNTH	21
C	ACTANT SPECIES (FOR ANY SPECIES NAME, THE FIRST *NSIZE* CHARACTERS	SYNTH	22
C	ARE RECOGNIZED, AND OTHERS IGNORED). EACH SIDE OF THE REACTION MAY	SYNTH	23
C	CONTAIN UP TO *MAX* SPECIES. THERE ARE NO RESTRICTIONS ON THE	SYNTH	24
C	REACTION FORMAT (WHICH MAY INCLUDE EMBEDDED BLANKS) EXCEPT FOR	SYNTH	25
C	THE FOLLOWING --	SYNTH	26
C		SYNTH	27
C	1) ELECTRONS ARE DENOTED BY EITHER E, E-, OR E(-).	SYNTH	28
C		SYNTH	29
C	2) HIGH ENERGY ELECTRONS (E-BEAM) ARE DENOTED BY HE-.	SYNTH	30
C		SYNTH	31
C	3) BUFFER GASES ARE DENOTED BY M.	SYNTH	32
C		SYNTH	33
C	4) NOISE PHOTONS ARE DENOTED BY HNU. IF HNU APPEARS ON THE	SYNTH	34
C	LEFT HAND SIDE, IT IS REJECTED. ALL PROCESSES WHICH CON-	SYNTH	35
C	TRIBUTE NOISE TO THE BUILDUP OF STIMULATED EMISSION ARE	SYNTH	36
C	TO BE DESCRIBED WITH HNU ON THE RIGHT HAND SIDE OF THE	SYNTH	37
C	REACTION.	SYNTH	38
C		SYNTH	39
C	5) STIMULATED EMISSION OR ABSORPTION PROCESSES ARE DESCRIBED	SYNTH	40
C	IN TERMS OF PHOTON NUMBER DENSITY, AND ARE RECOGNIZED BY	SYNTH	41
C	THE APPEARANCE OF RAD.	SYNTH	42
C		SYNTH	43
C	6) THERE ARE NO SPECIAL RESTRICTIONS ON SPECIES NAMES EXCEPT	SYNTH	44
C	THAT IONS MUST EXPLICITLY EXHIBIT THEIR CHARGE IN THEIR	SYNTH	45
C	NAME (E.G. F-, KR2(+), AR(+), CO-, ETC.). IF AN ION	SYNTH	46
C	IS POSITIVE, THE + SYMBOL MUST BE IMMEDIATELY FOLLOWED BY	SYNTH	47
C	EITHER ANOTHER + OR ), IN ORDER TO AVOID CONFUSION WITH	SYNTH	48
C	THE NORMAL USAGE OF + IN WRITING THE REACTION AS ABOVE.	SYNTH	49
C		SYNTH	50
C	IF KF = 0 (KR = 0), NO TRANSLATION OF THE FORWARD (REVERSE) RE-	SYNTH	51
C	ACTION TERM OCCURS (THUS, NULL OPERATIONS CONTAINING UNNECESSARY	SYNTH	52
C	MULTIPLICATIONS BY ZERO) ARE ELIMINATED. RATE CONSTANTS USED FOR	SYNTH	53
C	SYNTHESIS CAN BE CHANGED IN SUBSEQUENT EXECUTION, HOWEVER, SO A	SYNTH	54
C	NONZERO VALUE SHOULD BE USED DURING SYNTHESIS FOR ANY REACTION	SYNTH	55
C	PROCESS WHICH IS NOT TO BE PERMANENTLY NEGLECTED.	SYNTH	56
C		SYNTH	57
C	REACTIONS ARE SUBJECTED TO SEVERAL TESTS TO DETERMINE WHETHER THEY	SYNTH	58



C	SHOULD BE RETAINED FOR CONSTRUCTING THE KINETIC EQUATIONS IN THE	SYNTH	59
C	SUBROUTINE WHICH IS GENERATED. THESE INCLUDE THE FOLLOWING:	SYNTH	60
C		SYNTH	61
C	1) HIGH ENERGY ELECTRONS MUST BALANCE ON LHS AND RHS.	SYNTH	62
C		SYNTH	63
C	2) BUFFER GAS M MUST BALANCE ON BOTH SIDES OF EQUATION.	SYNTH	64
C		SYNTH	65
C	3) CHARGE CONSERVATION MUST NOT BE VIOLATED.	SYNTH	66
C		SYNTH	67
C	4) NO REVERSE PROCESS FOR SPONTANEOUS RADIATION ALLOWED.	SYNTH	68
C		SYNTH	69
C	5) DUPLICATE REACTIONS (EVEN WRITTEN BACKWARDS) ARE IGNORED	SYNTH	70
C		SYNTH	71
C	6) DETAIL BALANCE FOR BINARY MOLECULAR COLLISIONS ENFORCED.	SYNTH	72
C		SYNTH	73
C	7) SECONDARY E- COLLISION PROCESSES MAY HAVE FIXED RATE CON-	SYNTH	74
C	STANTS, OR ZERO MAY BE ENTERED, IN WHICH CASE THEY WILL BE	SYNTH	75
C	AUTOMATICALLY LINKED BY DEFAULT TO AN E- KINETICS ANALY-	SYNTH	76
C	SIS. DURING SYNTHESIS, THE ELECTRON CROSS SECTION FILE	SYNTH	77
C	WILL BE SCANNED TO DETERMINE WHETHER DATA FOR THE PROCESS	SYNTH	78
C	IS AVAILABLE, AND IF NOT, AN INFORMATIVE DIAGNOSTIC ISSUED	SYNTH	79
C		SYNTH	80
C	8) STIMULATED EMISSION (AND ABSORPTION) ARE DENOTED BY THE	SYNTH	81
C	SYNTAX: B + RAD + A + RAD. (IT IS ASSUMED THAT THE FOR-	SYNTH	82
C	WARD REACTION DENOTES EMISSION.) ABSORPTION PROCESSES CAN	SYNTH	83
C	BE ENTERED AS: X + RAD + Y. (ONLY THE FORWARD REACTION IS	SYNTH	84
C	RECOGNIZED, AND CORRESPONDS TO ABSORPTION.)	SYNTH	85
C		SYNTH	86
C	THERE ARE A VARIETY OF ERRORS RELATED TO EXCEEDED LIMITS, BAD	SYNTH	87
C	SYNTAX, BAD PHYSICS, ETC. WHICH ARE RECOGNIZED AND FLAGGED. BOTH	SYNTH	88
C	FATAL AND NON-FATAL WARNING CONDITIONS ARE GENERATED DURING SYNTH-	SYNTH	89
C	SIS, AND ARE PROVIDED IN AN EDITED OUTPUT SUMMARY OF THE REACTION	SYNTH	90
C	SCHEME WHICH WAS PROCESSED. A CROSS-REFERENCE LISTING OF THE	SYNTH	91
C	OCCURRENCE OF SPECIES IN THE REACTION SCHEME IS ALSO GENERATED.	SYNTH	92
C	THE SYNTHESIZED SUBROUTINES ARE DOCUMENTED WITH COMMENT CARDS.	SYNTH	93
C		SYNTH	94
C	.....	SYNTH	95
C		SYNTH	96
C	DIMENSION IMAGE(80), KAR(10), REFER(40), LINE(100), NBUFF(2),	SYNTH	97
C	1 NME(2), GAS(1), KINETIC(10), LABEL(5,2), LL(160), FORM(5),	SYNTH	98
C	2 LEV1(1), LEV2(1), VSIG(2), TITLE(3), NAME(100), COMM(5), KODE(8),	SYNTH	99
C	3 LDUM(10)	SYNTH	100
C		SYNTH	101
C	REAL KF, KR, KB	SYNTH	102
C		SYNTH	103
C	INTEGER GAS, SIGN, RATE(2), E, HNU, RAD, LHS, RHS, ME, TITLE, DATE	SYNTH	104
C		SYNTH	105
C	LOGICAL ELECT(2), RADIATE, REJECT, REVERSE, FORWARD, DETAIL, EXIT,	SYNTH	106
C	1 SOURCE, BUFFER, PHOTO(2), OPTICAL, LASER, TEST	SYNTH	107
C		SYNTH	108
C	DATA E, ME, HNU, RAD, F, R, SKIP / 4HE(-), 5HHE(-), 3HNU, 3HRAD,	SYNTH	109
C	1 7HFORWARD, 7HREVERSE, 5H(1H1) /	SYNTH	110
C		SYNTH	111
C	DATA KB, E0, H, C / 1.38 E-23, 1.602 E-19, 6.625 E-34, 3.0 E 10 /	SYNTH	112
C		SYNTH	113
C	NTAPE = FILE FOR GENERATION OF SUBROUTINES RATES, LEVELS	SYNTH	114
C	MTAPE = FILE FOR GENERATION OF SUBROUTINE JACOB	SYNTH	115



C	LTAPE = SCRATCH FILE FOR EDITING REACTION STRUCTURE	SYNTH	116
C	AT THE TERMINATION OF THE SYNTHESIS, NTAPE IS COPIED ONTO MTAPE.	SYNTH	117
C	NUATA = FILE CONTAINING UPDATED ELECTRON CROSS SECTION DATA	SYNTH	118
C	NSCRTCH = SCRATCH FILE FOR STORING REACTIONS AND RATE DATA	SYNTH	119
C		SYNTH	120
	KB = KB/E0	SYNTH	121
	EXIT = REJECT = .FALSE.	SYNTH	122
	IN = NTYPE = 2	SYNTH	123
	NAME(1) = GAS(1) = RAD	SYNTH	124
	NAME(2) = GAS(2) = E	SYNTH	125
	LASER = .FALSE.	SYNTH	126
	NPHOTON = 1	SYNTH	127
	READ (5,102) TITLE	SYNTH	128
	CALL COVER (TITLE,2)	SYNTH	129
	REWIND NSCRTCH	SYNTH	130
	WRITE (NSCRTCH) TITLE, DATE	SYNTH	131
C		SYNTH	132
C	GENERATE SYMBOLIC RATE SUBROUTINE TO BE EXECUTED --	SYNTH	133
C		SYNTH	134
	WRITE (NTAPE,400)	SYNTH	135
	WRITE (NTAPE,105)	SYNTH	136
	WRITE (NTAPE,406)	SYNTH	137
	WRITE (NTAPE,105)	SYNTH	138
	WRITE (NTAPE,605)	SYNTH	139
	WRITE (NTAPE,105)	SYNTH	140
	WRITE (NTAPE,460)	SYNTH	141
	WRITE (NTAPE,406)	SYNTH	142
	WRITE (NTAPE,105)	SYNTH	143
	WRITE (MTAPE,510)	SYNTH	144
	WRITE (MTAPE,105)	SYNTH	145
	WRITE (MTAPE,406)	SYNTH	146
	WRITE (MTAPE,105)	SYNTH	147
	WRITE (MTAPE,505)	SYNTH	148
	WRITE (MTAPE,105)	SYNTH	149
	WRITE (MTAPE,460)	SYNTH	150
	WRITE (MTAPE,406)	SYNTH	151
	WRITE (MTAPE,105)	SYNTH	152
	ENCODE (80,401,KODE)	SYNTH	153
	WRITE (NTAPE,102) KODE	SYNTH	154
	ENCODE (80,504,KODE)	SYNTH	155
	WRITE (MTAPE,102) KODE	SYNTH	156
	WRITE (NTAPE,402)	SYNTH	157
	WRITE (NTAPE,105)	SYNTH	158
	WRITE (NTAPE,104) KMAX, KMAX, KMAX, NKMAX, MAXGAS	SYNTH	159
	WRITE (NTAPE,512)	SYNTH	160
	WRITE (MTAPE,402)	SYNTH	161
	WRITE (MTAPE,105)	SYNTH	162
	WRITE (MTAPE,104) KMAX, KMAX, KMAX, NKMAX, MAXGAS	SYNTH	163
	WRITE (MTAPE,512)	SYNTH	164
	WRITE (NTAPE,421) KB, E0, H, C	SYNTH	165
	WRITE (MTAPE,421) KB, E0, H, C	SYNTH	166
C		SYNTH	167
	WRITE (NTAPE,403)	SYNTH	168
	WRITE (NTAPE,407)	SYNTH	169
	WRITE (NTAPE,105)	SYNTH	170
	WRITE (NTAPE,415)	SYNTH	171
	WRITE (NTAPE,105)	SYNTH	172

WRITE (NTAPE,422)	SYNTH	173
WRITE (NTAPE,406)	SYNTH	174
WRITE (NTAPE,105)	SYNTH	175
ENCODE (80,208,KODE)	SYNTH	176
WRITE (NTAPE,102) KODE	SYNTH	177
WRITE (NTAPE,105)	SYNTH	178
C	SYNTH	179
WRITE (MTAPE,403)	SYNTH	180
WRITE (MTAPE,507)	SYNTH	181
WRITE (MTAPE,105)	SYNTH	182
WRITE (MTAPE,422)	SYNTH	183
WRITE (MTAPE,406)	SYNTH	184
WRITE (MTAPE,105)	SYNTH	185
WRITE (MTAPE,102) KODE	SYNTH	186
WRITE (MTAPE,105)	SYNTH	187
C	SYNTH	188
C FORWARD DEFAULT FOR SECONDARY ELECTRON COLLISIONS --	SYNTH	189
M1 = 1	SYNTH	190
M2 = 2	SYNTH	191
INTEGER = 4H(I1)	SYNTH	192
INITIAL = 7HR = KF(	SYNTH	193
NTOT = 5H*NTOT	SYNTH	194
MULT = 4H*NO(	SYNTH	195
LAST = 6H - KR(	SYNTH	196
IF (NSIZE.GT.10) NSIZE = 10	SYNTH	197
MAX = 5	SYNTH	198
M0 = 5	SYNTH	199
M10 = 10*M0	SYNTH	200
M20 = 2*M10	SYNTH	201
LC = 0	SYNTH	202
NK = 0	SYNTH	203
INPUT = 0	SYNTH	204
K = 1	SYNTH	205
1 K = K-1	SYNTH	206
C READ HOLERITH STATEMENT OF REACTION NUMBER K --	SYNTH	207
10 K = K+1	SYNTH	208
READ (5,100) IMAGE	SYNTH	209
IF (EOF(5)) 5,76	SYNTH	210
76 READ (5,101) VSIG, COMM	SYNTH	211
DO 61 L = 1,M0	SYNTH	212
61 REFER(L) = COMM(L)	SYNTH	213
INPUT = INPUT+1	SYNTH	214
EXIT = EXIT.OR.REJECT	SYNTH	215
REWIND LTape	SYNTH	216
ENCODE (10,209,NUMBER) K	SYNTH	217
REJECT = DETAIL = .FALSE.	SYNTH	218
RATE(1) = RATE(2) = 1H	SYNTH	219
MESS = M0 + 1	SYNTH	220
IF (REFER(1).EQ.1H) MESS = 1	SYNTH	221
BUFFER = SOURCE = RADIATE = ELECT(1) = ELECT(2) = .FALSE.	SYNTH	222
PHOTO(1) = PHOTO(2) = .FALSE.	SYNTH	223
DO 9 L = 1,MAX	SYNTH	224
DO 9 M = 1,2	SYNTH	225
NBUFF(M) = NHE(M) = 0	SYNTH	226
9 LABEL(L,M) = 0	SYNTH	227
DO 22 L = 1,100	SYNTH	228
22 LINE(L) = 1H	SYNTH	229

DO 21 L = 1,160	SYNTH	230
21 LL(L) = 1H	SYNTH	231
NKAR = 6	SYNTH	232
DECODE (7,100,INITIAL) (LL(NKAR+L), L = 1,7)	SYNTH	233
NKAR = NKAR+7	SYNTH	234
K2 = K/10	SYNTH	235
K3 = K - 10*K2	SYNTH	236
K1 = K2/10	SYNTH	237
K2 = K2 - 10*K1	SYNTH	238
IF (K1.EQ.0) GO TO 11	SYNTH	239
NKAR = NKAR+1	SYNTH	240
ENCODE (10,INTEGER,LL(NKAR)) K1	SYNTH	241
11 IF (K1,K2.EQ.0) GO TO 29	SYNTH	242
NKAR = NKAR+1	SYNTH	243
ENCODE (10,INTEGER,LL(NKAR)) K2	SYNTH	244
29 NKAR = NKAR+1	SYNTH	245
ENCODE (10,INTEGER,LL(NKAR)) K3	SYNTH	246
NKAR = NKAR+1	SYNTH	247
LL(NKAR) = 1H	SYNTH	248
M = 1	SYNTH	249
N0 = NTYPE	SYNTH	250
NE = NCH = 0	SYNTH	251
NM = 1	SYNTH	252
LP = 1 = J = N = 0	SYNTH	253
C SCAN THE 80 BCD CHARACTERS TO DETERMINE SPECIES --	SYNTH	254
2 IF (I.EQ.80) GO TO 4	SYNTH	255
I = I+1	SYNTH	256
C IMBEDDED BLANKS ARE IGNORED.	SYNTH	257
IF (IMAGE(I).EQ.1H) GO TO 2	SYNTH	258
IF (IMAGE(I).NE.1H) GO TO 3	SYNTH	259
MM = 1	SYNTH	260
GO TO 4	SYNTH	261
C	SYNTH	262
3 IF (IMAGE(I).NE.1H) GO TO 6	SYNTH	263
IF (IMAGE(I+1).EQ.1H).OR.IMAGE(I+1).EQ.1H) GO TO 6	SYNTH	264
MM = 0	SYNTH	265
GO TO 4	SYNTH	266
C	SYNTH	267
C CONTINUE TO ADD NONBLANK BCD CHARACTERS TO THE GAS NAME. NAMES	SYNTH	268
C ARE TRUNCATED TO IGNORE ALL BUT THE FIRST NSIZE CHARACTERS.	SYNTH	269
C	SYNTH	270
6 IF (J.EQ.NSIZE) GO TO 2	SYNTH	271
J = J+1	SYNTH	272
LP = LP+1	SYNTH	273
LINE(LP) = KAR(J) = IMAGE(I)	SYNTH	274
IF (KAR(J).EQ.1H) NCH = NCH + NN	SYNTH	275
IF (KAR(J).EQ.1H-) NCH = NCH - NN	SYNTH	276
GO TO 2	SYNTH	277
C	SYNTH	278
C DUMP THE CONTENTS OF THE NAME AFTER A DELIMITER (*, %, OR COL. 80)	SYNTH	279
C HAS BEEN ENCOUNTERED --	SYNTH	280
C	SYNTH	281
4 IF (J.EQ.0) GO TO 60	SYNTH	282
IF (M.GT.2) GO TO 60	SYNTH	283
IF (I.EQ.80.AND.M.NE.2) GO TO 60	SYNTH	284
IF (N.LT.MAX) GO TO 38	SYNTH	285
ENCODE (M10,214,REFER(MESS)) MAX	SYNTH	286



MESS = MESS + M0	SYNTH	287
60 REJECT = .TRUE.	SYNTH	288
ENCODE (M10,216,REFER(MESS))	SYNTH	289
MESS = MESS+M0	SYNTH	290
LB = 0	SYNTH	291
DO 53 L = 1,80	SYNTH	292
IF (IMAGE(L).EQ.1H ) GO TO 53	SYNTH	293
LB = LB+1	SYNTH	294
IMAGE(LB) = IMAGE(L)	SYNTH	295
53 CONTINUE	SYNTH	296
LB = LB+1	SYNTH	297
IF (LB.GT.80) LB = 80	SYNTH	298
DO 55 L = LB,80	SYNTH	299
55 IMAGE(L) = 1H	SYNTH	300
NUMBER = 1H	SYNTH	301
C ENCODE UNRECOGNIZABLE REACTION --	SYNTH	302
ENCODE (100,100,KINETIC) (IMAGE(L), L = 1,80)	SYNTH	303
GO TO 26	SYNTH	304
38 ENCODE (10,100,NGAS) (KAR(L), L = 1,J)	SYNTH	305
IF (NGAS.EQ.1HE) NCH = NCH - NN	SYNTH	306
IF (NGAS.EQ.1HE.OR. NGAS.EQ.2HE-) NGAS = E	SYNTH	307
IF (NGAS.EQ.3HME-) NGAS = HE	SYNTH	308
IF (NGAS.EQ.HE) NCH = NCH + NN	SYNTH	309
J = 0	SYNTH	310
IF (NGAS.EQ.1HM) GO TO 8	SYNTH	311
IF (NGAS.EQ.HE) GO TO 8	SYNTH	312
IF (NGAS.NE.HNU) GO TO 30	SYNTH	313
IF (M.EQ.1) GO TO 67	SYNTH	314
IF (RADIATE) GO TO 67	SYNTH	315
IF (PHOTO(1).OR.PHOTO(2)) GO TO 67	SYNTH	316
RADIATE = .TRUE.	SYNTH	317
VSIG(2) = 0.	SYNTH	318
GO TO 16	SYNTH	319
30 IF (NGAS.NE.RAD) GO TO 45	SYNTH	320
IF (PHOTO(M)) GO TO 67	SYNTH	321
PHOTO(M) = .TRUE.	SYNTH	322
IF (RADIATE.AND.(PHOTO(1).OR.PHOTO(2))) GO TO 67	SYNTH	323
GO TO 45	SYNTH	324
67 REJECT = .TRUE.	SYNTH	325
ENCODE (M10,227,REFER(MESS))	SYNTH	326
MESS = MESS + M0	SYNTH	327
GO TO 16	SYNTH	328
45 N = N+1	SYNTH	329
IF (NGAS.EQ.E) NE = NE+1	SYNTH	330
DO 15 L = 1,N0	SYNTH	331
IF (GAS(L).NE.NGAS) GO TO 15	SYNTH	332
LABEL(N,M) = L	SYNTH	333
GO TO 8	SYNTH	334
15 CONTINUE	SYNTH	335
IF (N0.NE.MAXGAS) GO TO 39	SYNTH	336
IF (REJECT) GO TO 16	SYNTH	337
REJECT = .TRUE.	SYNTH	338
ENCODE (M20,213,REFER(MESS)) MAXGAS	SYNTH	339
MESS = MESS + 2*M0	SYNTH	340
GO TO 16	SYNTH	341
39 N0 = N0 + 1	SYNTH	342
GAS(N0) = NGAS	SYNTH	343

	LABEL(N,M) = N0	SYNTH	344
8	NEW = NKAR - 80*(NKAR/80) + 7	SYNTH	345
	IF (NEW.LE.72) GO TO 17	SYNTH	346
	NEW = NEW+8	SYNTH	347
	NKAR = 80*(NEW/80) + 6	SYNTH	348
	LL(NKAR) = 1H\$	SYNTH	349
17	DECODE (4,100,MULT) (LL(NKAR+L), L = 1,4)	SYNTH	350
	IF (NGAS.NE.1HM) GO TO 70	SYNTH	351
	NBUFF(M) = NBUFF(M) + 1	SYNTH	352
	BUFFER = .TRUE.	SYNTH	353
	DECODE (5,100,NTOT) (LL(NKAR+L), L = 1,5)	SYNTH	354
	NKAR = NKAR+5	SYNTH	355
	GO TO 16	SYNTH	356
70	IF (NGAS.NE.NE) GO TO 20	SYNTH	357
	NHE(M) = NHE(M) + 1	SYNTH	358
	SOURCE = .TRUE.	SYNTH	359
	GO TO 16	SYNTH	360
20	NKAR = NKAR+4	SYNTH	361
	N1 = LABEL(N,M)/10	SYNTH	362
	N2 = LABEL(N,M) - 10*N1	SYNTH	363
	IF (N1.EQ.0) GO TO 13	SYNTH	364
	NKAR = NKAR+1	SYNTH	365
	ENCODE (10,INTEGER,LL(NKAR)) N1	SYNTH	366
13	NKAR = NKAR+1	SYNTH	367
	ENCODE (10,INTEGER,LL(NKAR)) N2	SYNTH	368
	NKAR = NKAR+1	SYNTH	369
	LL(NKAR) = 1H	SYNTH	370
16	IF (1.EQ.80) GO TO 27	SYNTH	371
C		SYNTH	372
C	DELIMITERS (+, *) ENCODED INTO HOLERITH LINE TO DEFINE REACTION --	SYNTH	373
C		SYNTH	374
	LP = LP+1	SYNTH	375
	LINE(LP) = 1H	SYNTH	376
	LP = LP+1	SYNTH	377
	LINE(LP) = IMAGE(I)	SYNTH	378
	LP = LP+1	SYNTH	379
	LINE(LP) = 1H	SYNTH	380
	M = M + MM	SYNTH	381
	IF (MM.EQ.0) GO TO 2	SYNTH	382
	NL = NE	SYNTH	383
	IF (.NOT.SOURCE) GO TO 62	SYNTH	384
	IF (NHE(1).EQ.1) GO TO 62	SYNTH	385
	REJECT = .TRUE.	SYNTH	386
	ENCODE (M10,226,REFER(MESS))	SYNTH	387
	MESS = MESS + M0	SYNTH	388
62	IF (.NOT.(PHOTO(1).AND.NE.NE.0)) GO TO 64	SYNTH	389
	REJECT = .TRUE.	SYNTH	390
	ENCODE (M10,227,REFER(MESS))	SYNTH	391
	MESS = MESS + M0	SYNTH	392
64	ELECT(1) = .NOT.(PHOTO(1).OR.SOURCE.OR.BUFFER)	SYNTH	393
	DETAIL = .NOT.(PHOTO(1).OR.SOURCE)	SYNTH	394
	DETAIL = DETAIL.AND.(NE.EQ.0).AND.(N+NBUFF(1).EQ.2)	SYNTH	395
	ELECT(1) = (NE.EQ.1).AND.(N.EQ.2).AND.ELECT(1)	SYNTH	396
	IF (VSIG(1).NE.0.) ENCODE (10,103,RATE(1)) VSIG(1)	SYNTH	397
	FORWARD = (VSIG(1).NE.0.).AND.(NE.LE.1)	SYNTH	398
	FORWARD = FORWARD.OR.ELECT(1)	SYNTH	399
	LEFT = NKAR	SYNTH	400

	NN = - 1	SYNTH	401
	NE = N = 0	SYNTH	402
	IF (.NOT.FORWARD) NKAR = 9	SYNTH	403
C		SYNTH	404
C	CONSTRUCT REVERSE REACTION TERM --	SYNTH	405
	IF (REJECT) GO TO 2	SYNTH	406
	DECODE (6,100, LAST) (LL(NKAR+L), L = 1,6)	SYNTH	407
	NKAR = NKAR+6	SYNTH	408
	IF (K1.EQ.0) GO TO 24	SYNTH	409
	NKAR = NKAR+1	SYNTH	410
	ENCODE (10,INTEGER,LL(NKAR)) K1	SYNTH	411
24	IF (K1,K2.EQ.0) GO TO 28	SYNTH	412
	NKAR = NKAR+1	SYNTH	413
	ENCODE (10,INTEGER,LL(NKAR)) K2	SYNTH	414
28	NKAR = NKAR+1	SYNTH	415
	ENCODE (10,INTEGER,LL(NKAR)) K3	SYNTH	416
	NKAR = NKAR+1	SYNTH	417
	LL(NKAR) = 1H)	SYNTH	418
	GO TO 2	SYNTH	419
C		SYNTH	420
	27 IF (LP.GT.100) LP = 100	SYNTH	421
	ENCODE (100,100,KINETIC) (LINE(L), L = 1,LP)	SYNTH	422
	OPTICAL = PHOTO(1).OR.PHOTO(2)	SYNTH	423
C		SYNTH	424
	IF (NBUFF(1).EQ.NBUFF(2).AND.NBUFF(1).LE.1) GO TO 56	SYNTH	425
	ENCODE (M10,224,REFER(MESS))	SYNTH	426
	MESS = MESS + M0	SYNTH	427
	ENCODE (M10,216,REFER(MESS))	SYNTH	428
	MESS = MESS + M0	SYNTH	429
	REJECT = .TRUE.	SYNTH	430
	GO TO 23	SYNTH	431
C		SYNTH	432
	56 IF (NHE(1).EQ.NHE(2).AND.NHE(1).LE.1) GO TO 57	SYNTH	433
	ENCODE (M10,225,REFER(MESS))	SYNTH	434
	MESS = MESS + M0	SYNTH	435
	ENCODE (M10,216,REFER(MESS))	SYNTH	436
	MESS = MESS + M0	SYNTH	437
	REJECT = .TRUE.	SYNTH	438
	GO TO 23	SYNTH	439
C		SYNTH	440
	57 IF (.NOT.((SOURCE.AND.(RADIATE.OR.OPTICAL)).OR.(.NOT.PHOTO(1).AND.PHOTO(2)))) GO TO 66	SYNTH	441
1	ENCODE (M10,227,REFER(MESS))	SYNTH	442
	MESS = MESS + M0	SYNTH	443
	REJECT = .TRUE.	SYNTH	444
C		SYNTH	445
C	GENERATE CHECKSUM IDENTIFIER --	SYNTH	446
C		SYNTH	447
	66 IF (REJECT) GO TO 23	SYNTH	448
	K1 = K2 = NBUFF(1)*(MAXGAS+1) + NHE(1)*(MAXGAS+2)	SYNTH	449
	K1S0 = K2S0 = K1*K1 - 2*NBUFF(1)*(MAXGAS+1)*NHE(1)*(MAXGAS+2)	SYNTH	450
	DO 31 L = 1,5	SYNTH	451
	K1 = K1 + LABEL(L,1)	SYNTH	452
	K2 = K2 + LABEL(L,2)	SYNTH	453
	K1S0 = K1S0 + LABEL(L,1)*LABEL(L,1)	SYNTH	454
31	K2S0 = K2S0 + LABEL(L,2)*LABEL(L,2)	SYNTH	455
		SYNTH	456
		SYNTH	457



	ENCODE (10,107,LHS) K1, K1SQ	SYNTH	458
	ENCODE (10,107,RHS) K2, K2SQ	SYNTH	459
C		SYNTH	460
C	CHECK FOR DUPLICATION --	SYNTH	461
C		SYNTH	462
	IF (K.EQ.1) GO TO 23	SYNTH	463
	KM1 = K-1	SYNTH	464
	DO 25 L = 1,KM1	SYNTH	465
	READ (LTAPE) L1, L2	SYNTH	466
	IF (REJECT) GO TO 23	SYNTH	467
	REJECT = LHS.EQ.L1.AND.RHS.EQ.L2	SYNTH	468
	REVERSE = LHS.EQ.L2.AND.RHS.EQ.L1	SYNTH	469
	IF (RADIATE) REVERSE = .FALSE.	SYNTH	470
	REJECT = REJECT.OR.REVERSE	SYNTH	471
	IF (.NOT.REJECT) GO TO 25	SYNTH	472
	ENCODE (M10,210,REFER(MESS)) L	SYNTH	473
	IF (REVERSE) ENCODE (M10,211,REFER(MESS)) L	SYNTH	474
	MESS = MESS + M0	SYNTH	475
	25 CONTINUE	SYNTH	476
C		SYNTH	477
	23 ELECT(2) = (NE.EQ.1).AND.(N.EQ.2)	SYNTH	478
	NR = NE	SYNTH	479
	REVERSE = .NOT.(RADIATE.OR.(PHOTO(1).AND.VSIG(1).EQ.0.))	SYNTH	480
	IF (.NOT.REVERSE) VSIG(2) = 0.	SYNTH	481
	IF (PHOTO(1).AND..NOT.PHOTO(2)) VSIG(2) = 0.	SYNTH	482
	ELECT(2) = .NOT.(BUFFER.OR.RADIATE.OR.OPTICAL.	SYNTH	483
	1 OR.SOURCE).AND.ELECT(2)	SYNTH	484
	IF (VSIG(2).NE.0.) ENCODE (10,103,RATE(2)) VSIG(2)	SYNTH	485
	IF (VSIG(1).VSIG(2).NE.0.) ELECT(1) = ELECT(2) = .FALSE.	SYNTH	486
	IF (ELECT(1).AND.ELECT(2)) VSIG(1) = VSIG(2) = 0.	SYNTH	487
	ELECT(1) = ELECT(1).AND.VSIG(1).EQ.0.	SYNTH	488
	ELECT(2) = ELECT(2).AND.VSIG(2).EQ.0.	SYNTH	489
	DETAIL = (VSIG(1).NE.0.).AND.(VSIG(2).EQ.0.).AND.(NE.EQ.0.).AND.	SYNTH	490
	1 (N*NBUFF(2).EQ.2).AND.DETAIL.AND.REVERSE	SYNTH	491
	REVERSE = REVERSE.AND.(DETAIL.OR.(VSIG(2).NE.0.).OR.ELECT(2))	SYNTH	492
	REJECT = .NOT.(FORWARD.OR.REVERSE).OR.REJECT	SYNTH	493
	REJECT = REJECT.OR.(K.GT.KMAX)	SYNTH	494
	IF (K.LE.KMAX) GO TO 43	SYNTH	495
	ENCODE (M10,215,REFER(MESS)) KMAX	SYNTH	496
	MESS = MESS + M0	SYNTH	497
C		SYNTH	498
C	TEST FOR CHARGE CONSERVATION --	SYNTH	499
C		SYNTH	500
	43 IF (NCH.EQ.0) GO TO 85	SYNTH	501
	REJECT = .TRUE.	SYNTH	502
	ENCODE (M10,204,REFER(MESS))	SYNTH	503
	MESS = MESS + M0	SYNTH	504
	85 IF (REJECT) GO TO 51	SYNTH	505
	IF (.NOT.(ELECT(1).OR.ELECT(2))) GO TO 51	SYNTH	506
	IF (NK.LT.NKMAX) GO TO 48	SYNTH	507
	REJECT = .TRUE.	SYNTH	508
	ENCODE (M20,218,REFER(MESS)) NKMAX	SYNTH	509
	MESS = MESS + 2*M0	SYNTH	510
	GO TO 51	SYNTH	511
C		SYNTH	512
C	SEARCH FILE OF ELECTRON CROSS SECTIONS --	SYNTH	513
C		SYNTH	514

48 CALL DEKODE (NAME, IMAGE, L1, L2, LDUM, KAR, 10, IN, 60)	SYNTH	515
REWIND NDATA	SYNTH	516
74 READ (NDATA,100) IMAGE	SYNTH	517
IF (EOF(NDATA)) 79,72	SYNTH	518
72 CALL DEKODE (NAME, IMAGE, LL1, LL2, LDUM, KAR, 10, IN, 60)	SYNTH	519
IF (L1,NE,LL1.OR.L2,NE,LL2) GO TO 77	SYNTH	520
M1 = 1	SYNTH	521
M2 = 2	SYNTH	522
GO TO 52	SYNTH	523
C	SYNTH	524
77 IF (L1,NE,LL2.OR.L2,NE,LL1) GO TO 78	SYNTH	525
M1 = 2	SYNTH	526
M2 = 1	SYNTH	527
GO TO 52	SYNTH	528
C	SYNTH	529
C EXHAUST NUMERICAL DATA FOR THIS PROCESS --	SYNTH	530
C	SYNTH	531
78 READ (NDATA,100)	SYNTH	532
73 READ (NDATA,102) LETTERS	SYNTH	533
IF (LETTERS.EQ.1H) GO TO 74	SYNTH	534
READ (NDATA,102)	SYNTH	535
GO TO 73	SYNTH	536
C	SYNTH	537
79 ENCODE (M10,219,REFER(MESS))	SYNTH	538
MESS = MESS + M0	SYNTH	539
GO TO 52	SYNTH	540
C IF DESIRED, LACK OF E- CROSS SECTIONS CAN BE MADE SUFFICIENT FOR	SYNTH	541
C REJECTION OF THE REACTION BY THE REMOVAL OF THE ABOVE CARD, AND	SYNTH	542
C THE REMOVAL OF *C* ON THE FOLLOWING THREE CARDS --	SYNTH	543
C ELECT(1) = ELECT(2) = .FALSE.	SYNTH	544
C REJECT = .TRUE.	SYNTH	545
C GO TO 51	SYNTH	546
52 READ (NDATA,102) (REFER(L), L = 1,M0)	SYNTH	547
NK = NK+1	SYNTH	548
C KOLL(NK) = K	SYNTH	549
IF (.NOT.ELECT(1)) GO TO 54	SYNTH	550
RATE(1) = 10H(COMPUTED)	SYNTH	551
LEV1(NK) = K1-2	SYNTH	552
ENCODE (M10,223,REFER(MESS)) F	SYNTH	553
MESS = MESS + M0	SYNTH	554
54 IF (.NOT.ELECT(2)) GO TO 51	SYNTH	555
RATE(2) = 10H(COMPUTED)	SYNTH	556
LEV2(NK) = K2-2	SYNTH	557
ENCODE (M10,223,REFER(MESS)) R	SYNTH	558
MESS = MESS + M0	SYNTH	559
C	SYNTH	560
51 IF (NL,EQ.NR) GO TO 98	SYNTH	561
NE = NR-NL	SYNTH	562
IF (ELECT(1).AND.NE.LT.0) GO TO 98	SYNTH	563
IF (SOURCE) ENCODE (M10,312,REFER(MESS))	SYNTH	564
IF (.NOT.SOURCE) ENCODE (M10,313,REFER(MESS))	SYNTH	565
MESS = MESS + M0	SYNTH	566
98 IF (REJECT) NUMBER = 1H	SYNTH	567
IF (FORWARD) GO TO 46	SYNTH	568
ENCODE (M10,220,REFER(MESS))	SYNTH	569
MESS = MESS + M0	SYNTH	570
46 IF (REVERSE) GO TO 47	SYNTH	571

IF (.NOT.FORWARD) GO TO 49	SYNTH	572
ENCODE (M10,222,REFER(MESS))	SYNTH	573
MESS = MESS + M0	SYNTH	574
IF (.NOT.RADIATE) GO TO 26	SYNTH	575
ENCODE (M10,221,REFER(MESS))	SYNTH	576
MESS = MESS + M0	SYNTH	577
GO TO 26	SYNTH	578
49 MESS = MESS-M0	SYNTH	579
ENCODE (M10,212,REFER(MESS))	SYNTH	580
MESS = MESS + M0	SYNTH	581
IF (.NOT.RADIATE) GO TO 47	SYNTH	582
ENCODE (M10,221,REFER(MESS))	SYNTH	583
MESS = MESS + M0	SYNTH	584
GO TO 26	SYNTH	585
47 IF (.NOT.DETAIL) GO TO 26	SYNTH	586
RATE(2) = 10HX E(-E/KT)	SYNTH	587
ENCODE (M10,203,REFER(MESS))	SYNTH	588
MESS = MESS + M0	SYNTH	589
26 IF (LC.GT.0) GO TO 7	SYNTH	590
IF (K.GT.1) WRITE (6,202)	SYNTH	591
IF (K.GT.1) WRITE (6,470) DATE	SYNTH	592
WRITE (6,200)	SYNTH	593
7 IF (MESS.EQ.1) MESS = M0+1	SYNTH	594
REFER(MESS) = 1H	SYNTH	595
LC = LC+1*(MESS-1)/M0	SYNTH	596
IF (LC.GT.40) LC = 0	SYNTH	597
IF (REJECT.AND.VSIG(1).EQ.0.) RATE(1) = 1H	SYNTH	598
IF (REJECT.AND.VSIG(2).EQ.0.) RATE(2) = 1H	SYNTH	599
C THE REACTION IS STORED IN LINE(L), WITH BLANKS NEATLY EMBEDDED --	SYNTH	600
WRITE (6,201) NUMBER, (LINE(L), L = 1,45), RATE, (REFER(L), L =	SYNTH	601
1 1,MESS)	SYNTH	602
IF (REJECT) GO TO 1	SYNTH	603
NTYPE = NO	SYNTH	604
KTYPE = K	SYNTH	605
IF (ELECT(1)) ENCODE (10,110,RATE(1)) M1,NK	SYNTH	606
IF (ELECT(2)) ENCODE (10,110,RATE(2)) M2,NK	SYNTH	607
KF = KR = UNDEF	SYNTH	608
IF (FORWARD.AND.VSIG(1).NE.0.) KF = VSIG(1)	SYNTH	609
IF (REVERSE.AND.VSIG(2).NE.0.) KR = VSIG(2)	SYNTH	610
WRITE (LTAPE) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, (REFER(L),	SYNTH	611
1 L = 1,5)	SYNTH	612
PHOTO(2) = PHOTO(2).AND.PHOTO(1)	SYNTH	613
C	SYNTH	614
C	SYNTH	615
C	SYNTH	616
GENERATE RATE EXPRESSIONS --	SYNTH	617
IF (.NOT.REVERSE) NKAR = LEFT	SYNTH	618
WRITE (NTAPE,404) K, (KINETIC(L), L = 1,6)	SYNTH	619
WRITE (NTAPE,105)	SYNTH	620
WRITE (MTAPE,404) K, (KINETIC(L), L = 1,6)	SYNTH	621
WRITE (MTAPE,105)	SYNTH	622
IF (ELECT(1)) WRITE (NTAPE,405) K, M1, NK	SYNTH	623
IF (ELECT(1)) WRITE (MTAPE,405) K, M1, NK	SYNTH	624
IF (ELECT(2)) WRITE (NTAPE,409) K, M2, NK	SYNTH	625
IF (ELECT(2)) WRITE (MTAPE,409) K, M2, NK	SYNTH	626
IF (ELECT(1).OR.ELECT(2)) WRITE (NTAPE,105)	SYNTH	627
IF (ELECT(1).OR.ELECT(2)) WRITE (MTAPE,105)	SYNTH	628
IF (SOURCE) GO TO 65		



C	IF (.NOT.DETAIL) GO TO 44	SYNTH	629
	WRITE EXPRESSIONS FOR DETAILED BALANCE RELATION --	SYNTH	630
	WRITE (NTAPE,408)	SYNTH	631
	WRITE (NTAPE,105)	SYNTH	632
	WRITE (MTAPE,408)	SYNTH	633
	WRITE (MTAPE,105)	SYNTH	634
	GO TO 65	SYNTH	635
44	IF (.NOT.PHOTO(1)) GO TO 35	SYNTH	636
	IF (PHOTO(2)) WRITE (NTAPE,412) NPHOTON	SYNTH	637
	IF (.NOT.PHOTO(2)) WRITE (NTAPE,413) NPHOTON	SYNTH	638
	IF (PHOTO(2)) WRITE (MTAPE,412) NPHOTON	SYNTH	639
	IF (.NOT.PHOTO(2)) WRITE (MTAPE,413) NPHOTON	SYNTH	640
	IF (.NOT.PHOTO(2)) GO TO 35	SYNTH	641
	IF (LASER) GO TO 35	SYNTH	642
	LASER = .TRUE.	SYNTH	643
65	DO 41 M = 1,2	SYNTH	644
	ENCODE (10,106,LINE) M	SYNTH	645
	DO 42 L = 1,5	SYNTH	646
	J = LABEL(L,M)	SYNTH	647
	IF (J.EQ.0) GO TO 92	SYNTH	648
	L1 = L+1	SYNTH	649
	ENCODE (10,108,LINE(L1)) J	SYNTH	650
42	CONTINUE	SYNTH	651
92	IF (DETAIL) WRITE (MTAPE,109) (LINE(N), N = 1,L)	SYNTH	652
41	WRITE (NTAPE,109) (LINE(N), N = 1,L)	SYNTH	653
	IF (SOURCE) GO TO 35	SYNTH	654
	IF (DETAIL) GO TO 58	SYNTH	655
	WRITE (NTAPE,417)	SYNTH	656
	WRITE (NTAPE,105)	SYNTH	657
	GO TO 35	SYNTH	658
58	WRITE (NTAPE,410) K, K	SYNTH	659
	WRITE (NTAPE,105)	SYNTH	660
	WRITE (MTAPE,410) K, K	SYNTH	661
	WRITE (MTAPE,105)	SYNTH	662
35	WRITE (NTAPE,100) (LL(L), L = 1,NKAR)	SYNTH	663
	IF (SOURCE) WRITE (NTAPE,311)	SYNTH	664
	IF (PHOTO(1)) WRITE (NTAPE,416)	SYNTH	665
	WRITE (NTAPE,105)	SYNTH	666
	WRITE (NTAPE,217) K	SYNTH	667
	DO 18 I = 1,NTYPE	SYNTH	668
	N1 = N2 = 0	SYNTH	669
	DO 14 L = 1,MAX	SYNTH	670
	IF (LABEL(L,1).EQ.1) N1 = N1+1	SYNTH	671
14	IF (LABEL(L,2).EQ.1) N2 = N2+1	SYNTH	672
	N = N2-N1	SYNTH	673
C		SYNTH	674
C	IT IS UNDERSTOOD THAT, IF RAD APPEARS ON BOTH SIDES OF THE EQUA-	SYNTH	675
C	TION, A NET INCREASE IN RADIATION RESULTS.)	SYNTH	676
C		SYNTH	677
	IF (1.EQ.NPHOTON.AND.N1.EQ.1.AND.N2.EQ.1) N = 1	SYNTH	678
	IF (N.EQ.0) GO TO 18	SYNTH	679
	IF (N.GT.0) SIGN = 1H+	SYNTH	680
	IF (N.LT.0) SIGN = 1H-	SYNTH	681
	ENCODE (10,205,NSIGN) SIGN	SYNTH	682
	N = IABS(N)	SYNTH	683
	IND = 1	SYNTH	684
	IF (1.GT.9) IND = 2	SYNTH	685

IF (I.GT.99) IND = 3	SYNTH	686
ENCODE (50,500,FORM) IND, IND, MAX	SYNTH	687
WRITE (NTAPE,FORM) I, I, (NSIGN, J = 1,N)	SYNTH	688
18 CONTINUE	SYNTH	689
C IF (.NOT.SOURCE) GO TO 95	SYNTH	690
WRITE (NTAPE,305)	SYNTH	691
C DETERMINE IF SOURCE TERM CORRESPONDS TO IONIZATION --	SYNTH	692
DO 94 L = 1,MAX	SYNTH	693
J = LREL(L,2)	SYNTH	694
IF (J.EQ.0) GO TO 97	SYNTH	695
IF (J.EQ.2) GO TO 96	SYNTH	696
94 CONTINUE	SYNTH	697
C SOURCE TERM WAS IONIZATION --	SYNTH	698
96 WRITE (NTAPE,310)	SYNTH	699
97 WRITE (NTAPE,105)	SYNTH	700
WRITE (NTAPE,309)	SYNTH	701
WRITE (NTAPE,304)	SYNTH	702
GO TO 90	SYNTH	703
95 NE = NR-NL	SYNTH	704
IF (ELECT(1).OR.ELECT(2).OR.(NE.EQ.0)) GO TO 90	SYNTH	705
WRITE (NTAPE,105)	SYNTH	706
IF (NE.GT.0) SIGN = 1H+	SYNTH	707
IF (NE.LT.0) SIGN = 1H-	SYNTH	708
IF (NE.GT.0) WRITE (NTAPE,424)	SYNTH	709
IF (NE.LT.0) WRITE (NTAPE,425)	SYNTH	710
WRITE (NTAPE,105)	SYNTH	711
NE = IABS(NE)	SYNTH	712
WRITE (NTAPE,426) (SIGN, L = 1,NE)	SYNTH	713
90 IF (RADIATE) WRITE (NTAPE,418)	SYNTH	714
C .....	SYNTH	715
C	SYNTH	716
C	SYNTH	717
C GENERATE JACOBIAN --	SYNTH	718
C	SYNTH	719
C THE RATE FOR REACTION K, ENCODED INTO THE VECTOR LL ABOVE, IS R.	SYNTH	720
C IF SPECIES I OCCURS NR TIMES ON THE RIGHT, NL TIMES ON THE LEFT,	SYNTH	721
C OF REACTION K, THEN THE CONTRIBUTION OF REACTION K TO NDOT(I) IS	SYNTH	722
C NI*R, WHERE N = (NR - NL). THE FOLLOWING LOOP CALCULATES THE	SYNTH	723
C DERIVATIVE S = DS/DN(J) FOR EVERY SPECIES J WHICH OCCURS. FOR	SYNTH	724
C EVERY SPECIES I, REACTION K MAKES A CONTRIBUTION OF NI*S TO THE	SYNTH	725
C JACOBIAN PHI(I,J) = DNDOT(I)/DN(J).	SYNTH	726
C	SYNTH	727
C	SYNTH	728
C DO 89 J = 1,NTYPE	SYNTH	729
IP = JP = 0	SYNTH	730
LINE(1) = 1H(	SYNTH	731
NKAR = 1	SYNTH	732
M = JN = NJ = 0	SYNTH	733
83 M = M+1	SYNTH	734
IF (M.GT.2) GO TO 82	SYNTH	735
IF (M.EQ.2) GO TO 88	SYNTH	736
IF (.NOT.FORWARD) GO TO 83	SYNTH	737
NSIGN = 1H	SYNTH	738
KJ = 1HF	SYNTH	739
GO TO 84	SYNTH	740
88 IF (.NOT.REVERSE) GO TO 82	SYNTH	741
NSIGN = 1H-	SYNTH	742

C	KJ = 1HR	SYNTH	743
	NJ IS THE NUMBER OF TIMES SPECIES J APPEARS (ON LHS OR RHS) --	SYNTH	744
84	NJ = 0	SYNTH	745
	DO 63 L = 1,MAX	SYNTH	746
	JL = LABEL(L,M)	SYNTH	747
	IF (JL.NE.J) GO TO 63	SYNTH	748
	NJ = NJ+1	SYNTH	749
	LJ = L	SYNTH	750
63	CONTINUE	SYNTH	751
	JN = JN + NJ	SYNTH	752
	IF (NJ.EQ.0) GO TO 83	SYNTH	753
	IP = IP+1	SYNTH	754
	NKAR = NKAR+1	SYNTH	755
	LINE(NKAR) = NSIGN	SYNTH	756
	NKAR = NKAR+1	SYNTH	757
	ENCODE (10,306,LINE(NKAR)) KJ, K	SYNTH	758
	IF (NJ.GT.1) ENCODE (10,307,LINE(NKAR)) NJ, KJ, K	SYNTH	759
	DO 91 L = 1,MAX	SYNTH	760
	IF (L.EQ.LJ) GO TO 91	SYNTH	761
	JL = LABEL(L,M)	SYNTH	762
	IF (JL.EQ.0) GO TO 83	SYNTH	763
	NKAR = NKAR+1	SYNTH	764
	ENCODE (10,308,LINE(NKAR)) JL	SYNTH	765
91	CONTINUE	SYNTH	766
	GO TO 83	SYNTH	767
82	IF (JN.EQ.0) GO TO 89	SYNTH	768
	LKAR = NKAR = NKAR+1	SYNTH	769
	LINE(NKAR) = 1H	SYNTH	770
	IF (NHE(1).EQ.0) GO TO 68	SYNTH	771
	NKAR = NKAR+1	SYNTH	772
	LINE(NKAR) = 9H*IBEAM/E0	SYNTH	773
	JP = IP	SYNTH	774
68	IF (NBUFF(1).EQ.0) GO TO 69	SYNTH	775
	NKAR = NKAR+1	SYNTH	776
	LINE(NKAR) = NTOT	SYNTH	777
	JP = IP	SYNTH	778
69	IF (JP.NE.2) LINE(1) = LINE(LKAR) = 1H	SYNTH	779
	LKAR = 0	SYNTH	780
	DO 81 L = 1,NKAR	SYNTH	781
	DECODE (10,100,LINE(L)) (KAR(L)), L1 = 1,10)	SYNTH	782
	DO 93 L2 = 1,10	SYNTH	783
	IF (KAR(L2).EQ.1H) GO TO 93	SYNTH	784
	LKAR = LKAR+1	SYNTH	785
	LL(LKAR) = KAR(L2)	SYNTH	786
93	CONTINUE	SYNTH	787
81	CONTINUE	SYNTH	788
	WRITE (MTAPE,503) (LL(L), L = 1,LKAR)	SYNTH	789
	IF (PHOTO(1)) WRITE (MTAPE,416)	SYNTH	790
	WRITE (MTAPE,105)	SYNTH	791
C		SYNTH	792
	DO 86 I = 1,NTYPE	SYNTH	793
	N1 = N2 = 0	SYNTH	794
	DO 87 L = 1,MAX	SYNTH	795
	IF (LABEL(L,1).EQ.1) N1 = N1+1	SYNTH	796
87	IF (LABEL(L,2).EQ.1) N2 = N2+1	SYNTH	797
	N = N2-N1	SYNTH	798
	IF (I.EQ.NPHOTON.AND.N1.EQ.1.AND.N2.EQ.1) N = 1	SYNTH	799



IF (N.EQ.0) GO TO 86	SYNTH	800
IF (N.GT.0) SIGN = 1H+	SYNTH	801
IF (N.LT.0) SIGN = 1H-	SYNTH	802
ENCODE (10,205,NSIGN) SIGN	SYNTH	803
N = IABS(N)	SYNTH	804
WRITE (MTAPE,502) I, J, I, J, (NSIGN, L = 1,N)	SYNTH	805
IF (I.NE.1.OR.J.NE.1) GO TO 86	SYNTH	806
WRITE (MTAPE,105)	SYNTH	807
WRITE (MTAPE,503) (LL(L), L = 1,LKAR)	SYNTH	808
IF (PHOTO(2)) WRITE (MTAPE,411)	SYNTH	809
WRITE (MTAPE,420) NSIGN	SYNTH	810
86 CONTINUE	SYNTH	811
IF (RADIATE) WRITE (MTAPE,419) J, J	SYNTH	812
WRITE (MTAPE,105)	SYNTH	813
C	SYNTH	814
89 CONTINUE	SYNTH	815
C	SYNTH	816
WRITE (MTAPE,105)	SYNTH	817
GO TO 10	SYNTH	818
5 WRITE (MTAPE,423)	SYNTH	819
WRITE (6,202)	SYNTH	820
WRITE (6,470) DATE	SYNTH	821
NREJ = INPUT-KTYPE	SYNTH	822
IF (EXIT.AND.LC.GT.30) WRITE (6,SKIP)	SYNTH	823
WRITE (6,302) INPUT, KTYPE, KMAX, NREJ, NK, NTYPE, MAXGAS	SYNTH	824
IF (EXIT) WRITE (6,303)	SYNTH	825
C	SYNTH	826
C	SYNTH	827
C	SYNTH	828
EDIT THE REACTIONS TO DETERMINE WHERE EACH SPECIES OCCURS --	SYNTH	829
LC = 0	SYNTH	830
KOUNT = 40	SYNTH	831
TEST = .FALSE.	SYNTH	832
DO 50 I = 1,NTYPE	SYNTH	833
N = 0	SYNTH	834
REWIND LTape	SYNTH	835
DO 33 K = 1,KTYPE	SYNTH	836
READ (LTape) LHS, RHS, LABEL	SYNTH	837
DO 34 M = 1,2	SYNTH	838
DO 34 L = 1,5	SYNTH	839
IF (LABEL(L,M).EQ.I) GO TO 36	SYNTH	840
34 CONTINUE	SYNTH	841
GO TO 37	SYNTH	842
36 N = N+1	SYNTH	843
LINE(N) = K	SYNTH	844
37 IF (N.LT.160.AND.K.LT.KTYPE) GO TO 33	SYNTH	845
IF (N.EQ.0) GO TO 33	SYNTH	846
IF (LC.NE.0) GO TO 32	SYNTH	847
IF (TEST) WRITE (6,202)	SYNTH	848
IF (TEST) WRITE (6,470) DATE	SYNTH	849
TEST = .TRUE.	SYNTH	850
WRITE (6,300) NTYPE	SYNTH	851
WRITE (6,202)	SYNTH	852
32 LC = LC + 2 + N/20	SYNTH	853
WRITE (6,301) I, GAS(I), (LINE(L), L = 1,N)	SYNTH	854
IF (LC.GT.KOUNT) LC = 0	SYNTH	855
N = 0	SYNTH	856
33 CONTINUE		

50	CONTINUE	SYNTH	857
	WRITE (6,202)	SYNTH	858
	WRITE (6,470) DATE	SYNTH	859
C		SYNTH	860
	WRITE (NSCRTCH) MAXGAS, NTYPE, KMAX, KTYPE, NKMAX, NK, EXIT	SYNTH	861
	WRITE (NSCRTCH) (GAS(I), I = 1, NTYPE)	SYNTH	862
	WRITE (NSCRTCH) (LEV1(I), LEV2(I), I = 1, NK)	SYNTH	863
C		SYNTH	864
C	COPY LTape ONTO NSCRTCH --	SYNTH	865
	REWIND LTape	SYNTH	866
	DO 59 K = 1, KTYPE	SYNTH	867
	READ (LTape) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, COMM	SYNTH	868
	WRITE (NSCRTCH) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, COMM	SYNTH	869
59	CONTINUE	SYNTH	870
C		SYNTH	871
	NP1 = NTYPE + 1	SYNTH	872
	NP2 = NTYPE + 2	SYNTH	873
C		SYNTH	874
	WRITE (MTAPE,414)	SYNTH	875
	WRITE (MTAPE,105)	SYNTH	876
	WRITE (MTAPE,406)	SYNTH	877
	WRITE (MTAPE,406)	SYNTH	878
	WRITE (MTAPE,105)	SYNTH	879
	WRITE (MTAPE,105)	SYNTH	880
	ENCODE (80,506,KODE) NP1, NP2	SYNTH	881
	WRITE (MTAPE,102) KODE	SYNTH	882
	WRITE (MTAPE,102) KODE	SYNTH	883
	WRITE (MTAPE,105)	SYNTH	884
	WRITE (MTAPE,105)	SYNTH	885
	WRITE (MTAPE,508)	SYNTH	886
	WRITE (MTAPE,508)	SYNTH	887
	ENCODE (80,509,KODE) NP1, NP2	SYNTH	888
	WRITE (MTAPE,102) KODE	SYNTH	889
	ENCODE (80,513,KODE) NP2, NP1, NP2	SYNTH	890
	WRITE (MTAPE,102) KODE	SYNTH	891
	WRITE (MTAPE,440)	SYNTH	892
	WRITE (MTAPE,105)	SYNTH	893
	ENCODE (80,514,KODE) NP1, NP1	SYNTH	894
	WRITE (MTAPE,102) KODE	SYNTH	895
	ENCODE (80,515,KODE) NP2, NP1	SYNTH	896
	WRITE (MTAPE,102) KODE	SYNTH	897
	WRITE (MTAPE,105)	SYNTH	898
C		SYNTH	899
	ENCODE (80,511,KODE) NP1, NP1	SYNTH	900
	WRITE (MTAPE,102) KODE	SYNTH	901
	ENCODE (80,516,KODE) NP1, NP2	SYNTH	902
	WRITE (MTAPE,102) KODE	SYNTH	903
	ENCODE (80,517,KODE) NP2, NP1	SYNTH	904
	WRITE (MTAPE,102) KODE	SYNTH	905
	ENCODE (80,518,KODE) NP2, NP2	SYNTH	906
	WRITE (MTAPE,102) KODE	SYNTH	907
	ENCODE (80,519,KODE) NP2, NP2	SYNTH	908
	WRITE (MTAPE,102) KODE	SYNTH	909
	WRITE (MTAPE,440)	SYNTH	910
	WRITE (MTAPE,105)	SYNTH	911
	ENCODE (80,521,KODE) NP1, NP1	SYNTH	912
	WRITE (MTAPE,102) KODE	SYNTH	913

ENCODE (80,520,KODE) NP1, NP1	SYNTH	914
WRITE (MTAPE,102) KODE	SYNTH	915
WRITE (MTAPE,105)	SYNTH	916
C	SYNTH	917
WRITE (NTAPE,406)	SYNTH	918
WRITE (NTAPE,105)	SYNTH	919
WRITE (MTAPE,406)	SYNTH	920
WRITE (MTAPE,105)	SYNTH	921
ENCODE (80,207,KODE)	SYNTH	922
WRITE (NTAPE,102) KODE	SYNTH	923
WRITE (NTAPE,105)	SYNTH	924
WRITE (MTAPE,102) KODE	SYNTH	925
WRITE (MTAPE,105)	SYNTH	926
N2 = 0	SYNTH	927
12 N1 = N2+1	SYNTH	928
IF (N1.GT.NTYPE) GO TO 19	SYNTH	929
N2 = N1+3	SYNTH	930
IF (N2.GT.NTYPE) N2 = NTYPE	SYNTH	931
ENCODE (80,206,KODE) (L, GAS(L), L = N1,N2)	SYNTH	932
WRITE (NTAPE,102) KODE	SYNTH	933
WRITE (MTAPE,102) KODE	SYNTH	934
GO TO 12	SYNTH	935
19 WRITE (NTAPE,105)	SYNTH	936
WRITE (NTAPE,406)	SYNTH	937
WRITE (NTAPE,105)	SYNTH	938
WRITE (NTAPE,440)	SYNTH	939
WRITE (NTAPE,450)	SYNTH	940
WRITE (MTAPE,105)	SYNTH	941
WRITE (MTAPE,406)	SYNTH	942
WRITE (MTAPE,105)	SYNTH	943
WRITE (MTAPE,440)	SYNTH	944
WRITE (MTAPE,450)	SYNTH	945
C	SYNTH	946
C	SYNTH	947
C	SYNTH	948
C	SYNTH	949
CONSTRUCT SUBROUTINE TO COMPUTE POPULATION DENSITIES FOR THE	SYNTH	950
LOWER AND UPPER LEVELS TO BE USED IN THE E- KINETICS ANALYSIS --	SYNTH	951
IF (NK.EQ.0) GO TO 75	SYNTH	952
WRITE (NTAPE,600)	SYNTH	953
WRITE (NTAPE,105)	SYNTH	954
WRITE (NTAPE,406)	SYNTH	955
WRITE (NTAPE,105)	SYNTH	956
WRITE (NTAPE,604)	SYNTH	957
WRITE (NTAPE,105)	SYNTH	958
WRITE (NTAPE,460)	SYNTH	959
WRITE (NTAPE,406)	SYNTH	960
WRITE (NTAPE,105)	SYNTH	961
ENCODE (80,601,KODE)	SYNTH	962
WRITE (NTAPE,102) KODE	SYNTH	963
WRITE (NTAPE,105)	SYNTH	964
DO 80 I = 1,NK	SYNTH	965
ENCODE (80,602,KODE) M1, I, LEV1(I)	SYNTH	966
IF (LEV1(I).EQ.0) ENCODE (80,603,KODE) M1, I	SYNTH	967
WRITE (NTAPE,102) KODE	SYNTH	968
ENCODE (80,602,KODE) M2, I, LEV2(I)	SYNTH	969
IF (LEV2(I).EQ.0) ENCODE (80,603,KODE) M2, I	SYNTH	970
WRITE (NTAPE,102) KODE		
80 WRITE (NTAPE,105)		



	WRITE (NTAPE,440)	SYNTH	971
	WRITE (NTAPE,450)	SYNTH	972
C		SYNTH	973
C	COPY NTAPE ONTO MTAPE --	SYNTH	974
C		SYNTH	975
	75 REWIND NTAPE	SYNTH	976
	40 READ (NTAPE,102) KODE	SYNTH	977
	IF (EOF(NTAPE)) 99,71	SYNTH	978
	71 WRITE (MTAPE,102) KODE	SYNTH	979
	GO TO 40	SYNTH	980
	99 REWIND MTAPE	SYNTH	981
	REWIND NTAPE	SYNTH	982
C		SYNTH	983
C	----- FORMAT STATEMENTS -----	SYNTH	984
C		SYNTH	985
	100 FORMAT (80A1)	SYNTH	986
C		SYNTH	987
	101 FORMAT (2E10.3,5X,5A10)	SYNTH	988
C		SYNTH	989
	102 FORMAT (8A10)	SYNTH	990
C		SYNTH	991
	103 FORMAT (1PE10.4)	SYNTH	992
C		SYNTH	993
	104 FORMAT (6X*COMMON / DATA / RATE(*I4*), KF(*I4*), KR(*I4*), VSIG(2,	SYNTH	994
	1*I3*), E(*I2*)*8X)	SYNTH	995
C		SYNTH	996
	105 FORMAT (1HC,79X)	SYNTH	997
C		SYNTH	998
	106 FORMAT (6X*E*,11,* =*)	SYNTH	999
C		SYNTH	1000
	107 FORMAT (I4,I6)	SYNTH	1001
C		SYNTH	1002
	108 FORMAT (* + E(*,I2,*)*)	SYNTH	1003
C		SYNTH	1004
	109 FORMAT (A10,5A8,30X)	SYNTH	1005
C		SYNTH	1006
	110 FORMAT (*VSIG(*,11,*,*,I2,*)*)	SYNTH	1007
C		SYNTH	1008
	200 FORMAT (1H1/17X*SUMMARY OF INPUT: REACTIONS AND RATE CONSTANTS (S	SYNTH	1009
	1EC-1, CM3/SEC, CM6/SEC, ... OR CM2) WITH REFERENCES*///10X*(IF A	SYNTH	1010
	2RATE CONSTANT KF OR KR FOR A BINARY ELECTRON COLLISION IS NOT EXPL	SYNTH	1011
	3ICITLY SPECIFIED. IT WILL BE COMPUTED SELF*/10X*CONSISTENTLY AS A	SYNTH	1012
	4FUNCTION OF E/N, GAS COMPOSITION, AND EXCITED LEVEL DENSITIES FROM	SYNTH	1013
	5 A COUPLED ELECTRON ANALYSIS.)*///4X,*1*12X,*REACTION(I)*,34X,	SYNTH	1014
	6*RATE CONSTANTS*14X*RATE REFERENCES AND/OR COMMENTS*8X*(IGNORED R	SYNTH	1015
	7EACTIONS ARE NOT NUMBERED)*.16X,*KF(I)*8X,*KR(I)*1X,134(1H-)//)	SYNTH	1016
C		SYNTH	1017
	201 FORMAT (A8,45A1,4X,A10,3X,A10,4X,5A10/(84X,5A10))	SYNTH	1018
C		SYNTH	1019
	202 FORMAT (/1X,134(1H-))	SYNTH	1020
C		SYNTH	1021
	203 FORMAT (*REVERSE RATE IS OBTAINED FROM DETAILED BALANCE.*)	SYNTH	1022
C		SYNTH	1023
	204 FORMAT (*REACTION REJECTED -- CHARGE CONSERVATION VIOLATED.*)	SYNTH	1024
C		SYNTH	1025
	205 FORMAT (1H ,A1,* R*)	SYNTH	1026
C		SYNTH	1027

206	FORMAT (*C*4(I4,3X,A10))	SYNTH	1028
C		SYNTH	1029
207	FORMAT (*C THE FOLLOWING MOLECULAR SPECIES (WITH LABELS) WERE INCLUDED --*)	SYNTH	1030
		SYNTH	1031
C		SYNTH	1032
208	FORMAT (*C THE FOLLOWING REACTIONS DEFINE THE KINETICS --*)	SYNTH	1033
C		SYNTH	1034
209	FORMAT (I5)	SYNTH	1035
C		SYNTH	1036
210	FORMAT (*REACTION IGNORED -- SAME AS NUMBER*,I4,*,*)	SYNTH	1037
C		SYNTH	1038
211	FORMAT (*REACTION IGNORED -- REVERSE OF NO,*,I4,*,*)	SYNTH	1039
C		SYNTH	1040
212	FORMAT (*REACTION IS IGNORED -- KF = KR = 0.*)	SYNTH	1041
C		SYNTH	1042
213	FORMAT (*REACTION IS IGNORED -- MORE THAN*,I3,* DIFFERENT GAS SPECIES ARE NOT PERMITTED WITH PRESENT DIMENSION.*)	SYNTH	1043
		SYNTH	1044
C		SYNTH	1045
214	FORMAT (* *I1,* SPECIES ON LHS OR RHS NOT PERMITTED.*)	SYNTH	1046
C		SYNTH	1047
215	FORMAT (*MORE THAN *,I3,* REACTIONS ARE IGNORED.*)	SYNTH	1048
C		SYNTH	1049
216	FORMAT (*BAD SYNTAX -- UNRECOGNIZABLE REACTION IS IGNORED.*)	SYNTH	1050
C		SYNTH	1051
217	FORMAT (I1X*RATE(*I3*) = R*56X)	SYNTH	1052
C		SYNTH	1053
218	FORMAT (*REACTION REJECTED -- NO MORE THAN*,I3,* REACTIONS ALLOWED FOR THE COUPLED E- KINETICS ANALYSIS.*)	SYNTH	1054
		SYNTH	1055
C		SYNTH	1056
219	FORMAT (*WARNING -- NO E- CROSS SECTION DATA WAS FOUND.*)	SYNTH	1057
C		SYNTH	1058
220	FORMAT (*FORWARD REACTION IS IGNORED -- KF = 0.*)	SYNTH	1059
C		SYNTH	1060
221	FORMAT (*NO REVERSE REACTION ALLOWED FOR RADIATIVE DECAY.*)	SYNTH	1061
C		SYNTH	1062
222	FORMAT (*REVERSE REACTION IS IGNORED -- KR = 0.*)	SYNTH	1063
C		SYNTH	1064
223	FORMAT (A7,* RATE IS OBTAINED FROM E- KINETICS ANALYSIS*)	SYNTH	1065
C		SYNTH	1066
224	FORMAT (*IMPROPER BUFFER GAS SPECIFICATION.*)	SYNTH	1067
C		SYNTH	1068
225	FORMAT (*IMPROPER HIGH ENERGY ELECTRON TERMS.*)	SYNTH	1069
C		SYNTH	1070
226	FORMAT (*THREE-BODY HE- COLLISION NOT ALLOWED.*)	SYNTH	1071
C		SYNTH	1072
227	FORMAT (*THIS RADIATIVE PROCESS NOT ALLOWED.*)	SYNTH	1073
C		SYNTH	1074
300	FORMAT (I1I/35X,*SUMMARY OF REACTIONS FOR WHICH EACH SPECIES OCCUR IS: NTYPE = *,I3/31X*(THIS EDIT PERMITS RAPID DELETION OF ANY SPECIES FROM THE KINETIC SYSTEM)*/7X,*I5X*GAS(I)*40X*REACTIONS CONTAINING GAS(I)*)	SYNTH	1075
		SYNTH	1076
		SYNTH	1077
		SYNTH	1078
C		SYNTH	1079
301	FORMAT (/I8,5X,A10,5X,I4,I9(*,*,I4)/(28X,I4,I9(*,*,I4)))	SYNTH	1080
C		SYNTH	1081
302	FORMAT (I/4X,*OF*,I4,* INPUT REACTIONS SCANNED,*,I4,* WERE RETAINED I (MAXIMUM ALLOWED =*,I4,*) AND*,I4,* WERE IGNORED FOR REASONS ITEM SIZED IN THE TABLE*/4X,*OF THOSE RETAINED,*,I3,* REQUIRE RATES FROM	SYNTH	1082
		SYNTH	1083
		SYNTH	1084

3	AN E- KINETICS ANALYSIS. *,I3,* SEPARATE SPECIES WERE ENCOUNTERED	SYNTH	1085
	4 (MAXIMUM ALLOWED =*,I3,*).*/)	SYNTH	1086
C		SYNTH	1087
303	FORMAT (4X*ERRORS WHICH WERE DETECTED IN PROCESSING THE INPUT REAC	SYNTH	1088
	ITION SCHEME MAY CAUSE PROGRAM TERMINATION IF THEY HAVE BEEN SPECIF	SYNTH	1089
	2IED TO*/4X,*BE TREATED AS FATAL. MODIFICATIONS OF THE REACTION SC	SYNTH	1090
	HEME, CORRECTIONS IN REACTION SYNTAX, CHANGES IN DIMENSION STORAGE	SYNTH	1091
	4, OR*/4X,*ADDITIONS TO THE E- CROSS SECTION FILE MAY BE REQUIRED T	SYNTH	1092
	50 REMOVE ALL OF THE ERROR DECLARATIONS.*)	SYNTH	1093
C		SYNTH	1094
304	FORMAT (11X#DVDX = DVDX + U*S#52X)	SYNTH	1095
C		SYNTH	1096
305	FORMAT (11X*U = (E2 - E1)*.56X)	SYNTH	1097
C		SYNTH	1098
306	FORMAT (*K*.A1,*(*.I3,*).*)	SYNTH	1099
C		SYNTH	1100
307	FORMAT (11.3H.*K.A1,*(*.I3,*).*)	SYNTH	1101
C		SYNTH	1102
308	FORMAT (4H*NO(.I2,*).*)	SYNTH	1103
C		SYNTH	1104
309	FORMAT (*C*5X*E-BEAM ENERGY DEPOSITION --*47X)	SYNTH	1105
C		SYNTH	1106
310	FORMAT (*C*79X/*C*5X*SECONDARY ELECTRON CREATION--*45X/11X*SB = SB	SYNTH	1107
	1 + R*58X/11X*U = U + UPLUS*56X)	SYNTH	1108
C		SYNTH	1109
311	FORMAT (6X*S = R*69X/6X#R = R*IBEAM/E0#60X)	SYNTH	1110
C		SYNTH	1111
312	FORMAT (*E- CREATION ASSUMED TO BE OVER ENERGY DISTRIBUTION*)	SYNTH	1112
C		SYNTH	1113
313	FORMAT (*E- CREATED (OR LOST) ASSUMED TO BE AT ZERO ENERGY*)	SYNTH	1114
C		SYNTH	1115
400	FORMAT (#*DECK,DNDT#.70X/6X*SURROUTINE DNDT (N. T. NO. NDOT)*42X)	SYNTH	1116
C		SYNTH	1117
401	FORMAT (6X*DIMENSION NO(1), NDOT(1)*50X)	SYNTH	1118
C		SYNTH	1119
402	FORMAT (6X*REAL NO, NTOT, NDOT, NOISE, NE, KF, KR, KB, KT, MU, L0,	SYNTH	1120
	1 IBEAM,*12X/5X*1 JBEAM, LENGTH*60X)	SYNTH	1121
C		SYNTH	1122
403	FORMAT (6X#IBEAM = JBEAM*DEPOSIT*SHAPE(T)#44X/6X*KT = KB*TMOL#62X)	SYNTH	1123
C		SYNTH	1124
404	FORMAT (1HC,I3.2X.6A10.5X)	SYNTH	1125
C		SYNTH	1126
405	FORMAT (*C FORWARD RATE IS OBTAINED FROM E(-) KINETICS ANALYSI	SYNTH	1127
	IS*.22X/11X,*KF(*.I3,*).*) = VSIG(*.I1,*.*.I2,*).*.48X)	SYNTH	1128
C		SYNTH	1129
406	FORMAT (*C *.69(1H.).8X)	SYNTH	1130
C		SYNTH	1131
407	FORMAT (6X*DO 1 I = 1,N*62X/4X*1 NDOT(I) = 0.*62X/6X#NDOT(1) = - C	SYNTH	1132
	1*GAMMA*NO(1)#49X/6X*ALPHA = GAIN = HNU = FREQ = NOISE = DVDX = 0.*	SYNTH	1133
	229X/6X*SB = 50 = 0.0*61X)	SYNTH	1134
C		SYNTH	1135
408	FORMAT (*C*.5X.*REVERSE RATE IS OBTAINED FROM DETAIL BALANCE --*,	SYNTH	1136
	127X)	SYNTH	1137
C		SYNTH	1138
409	FORMAT (*C REVERSE RATE IS OBTAINED FROM E(-) KINETICS ANALYSI	SYNTH	1139
	IS*.22X/11X,*KR(*.I3,*).*) = VSIG(*.I1,*.*.I2,*).*.48X)	SYNTH	1140
C		SYNTH	1141



C	410	FORMAT (*	KR(*,I3,*) = KF(*,I3,I8H)*EXP(-(E1-E2)/KT),40X)	SYNTH	1142
C	411	FORMAT (11X*GAIN = GAIN + R*54X)		SYNTH	1143
C	412	FORMAT (*C*5X*(STIMULATED EMISSION PROCESS, WITH NO(*I,*) = INTEN/ 1C/HNU)*17X/*C*79X)		SYNTH	1144
C	413	FORMAT (*C*5X*(RADIATIVE ABSORPTION PROCESS, WITH NO(*I2*) = INTEN/ 1C/HNU)*16X/*C*79X)		SYNTH	1145
C	414	FORMAT (*C*5X*COMPUTE FINAL EXPRESSION FOR D/DT(PHOTON DENSITY) -- 1*22X/*C*79X/6X#NDOT(1) = (LENGTH/CAVITY)*(NDOT(1) + NOISE)*31X/*C* 279X/6X*ABSORB = GAIN - ALPHA*53X/6X,#DVDX = DEPOSIT*DVDX*55X)		SYNTH	1146
C	415	FORMAT (*C*5X*GAIN = SIGMA*(N2-N1) IS THE LASER TRANSITION GAIN# 1 23X/*C*5X*ABSORB = SUMK(SIGMA(K)*NK) IS THE TOTAL ABSORPTION OF T 2HE MEDIUM#10X/*C*5X*ALPHA = (GAIN-ABSORB) IS THE NET GAIN IN THE 3MEDIUM*22X/*C*5X*GAMMA = THRESHOLD GAIN COEFFICIENT (CM-1)*31X)		SYNTH	1147
C	416	FORMAT (6X#R = R*C*67X)		SYNTH	1148
C	417	FORMAT (6X#HNU = E0*(E1 - E2)*56X/6X*FREQ = HNU/H*62X)		SYNTH	1149
C	418	FORMAT (*C*79X/*C*5X*PHOTON NUMBER DENSITY INCREASED BY NOISE -- 131X/*C*79X/11X#NOISE = NOISE + R*OMEGA/4./PI*40X)		SYNTH	1150
C	419	FORMAT (*C*79X/*C*5X*PHOTON NUMBER DENSITY INCREASED BY NOISE -- 131X/*C*79X/11X#R = R*OMEGA/4./PI*51X/11X*PHI(1.*I2*) = PHI(1.*I2 2*) + R*44X)		SYNTH	1151
C	420	FORMAT (11X*ALPHA = ALPHA*A4.52X)		SYNTH	1152
C	421	FORMAT (6X*DATA KB. E0, H, C, PI /*IPE10.3*,*IPE10.3*,*IPE10.3*,* 118X/5X*1 *IPE10.3*, 3.14159 /*49X/*C*79X)		SYNTH	1153
C	422	FORMAT (*C*5X*CAVITY = MIRROR SEPARATION (CM)*44X/*C*5X*LENGTH = L LENGTH OF ACTIVE MEDIUM (CM)*37X/*C*5X*OMEGA = AREA/CAVITY*2*51X/ 2*C*5X*AREA = AREA OF OPTICS (CM2)*45X/*C*5X*GAMMA = [LOSS + LN( 31/R)/2]/LENGTH*40X/*C*79X)		SYNTH	1154
C	423	FORMAT (6X*RATIO = LENGTH/CAVITY*53X/6X*DO 3 I = 1,N*62X/4X#3 PHI( 11,I) = RATIO*PHI(1,I)*49X/*C*79X)		SYNTH	1155
C	424	FORMAT (*C*5X*CREATION OF (ZERO ENERGY) SECONDARY ELECTRONS--*27X)		SYNTH	1156
C	425	FORMAT (*C*5X*LOSS OF (ZERO ENERGY) SECONDARY ELECTRONS --*30X)		SYNTH	1157
C	426	FORMAT (11X*S0 = S0 *A1* R*58X)		SYNTH	1158
C	440	FORMAT (6X*RETURN*68X)		SYNTH	1159
C	450	FORMAT (6X*END*71X)		SYNTH	1160
C	460	FORMAT (*C*5X*THE GENERAL KINETICS SYNTHESIS PROGRAM WHICH AUTOMAT 1ICALLY GEN-11X/*C*5X*ERATED THIS SUBROUTINE WAS DEVELOPED BY -- 232X/*C*79X/*C*15X,40(1H-),24X/*C*15X*I*38X*I*24X/*C*15X*I DR. WI 3LLIAM B. LACINA*14X*I*24X/*C*15X*I NORTHROP RESEARCH AND TECHNOL		SYNTH	1161
				SYNTH	1162
				SYNTH	1163
				SYNTH	1164
				SYNTH	1165
				SYNTH	1166
				SYNTH	1167
				SYNTH	1168
				SYNTH	1169
				SYNTH	1170
				SYNTH	1171
				SYNTH	1172
				SYNTH	1173
				SYNTH	1174
				SYNTH	1175
				SYNTH	1176
				SYNTH	1177
				SYNTH	1178
				SYNTH	1179
				SYNTH	1180
				SYNTH	1181
				SYNTH	1182
				SYNTH	1183
				SYNTH	1184
				SYNTH	1185
				SYNTH	1186
				SYNTH	1187
				SYNTH	1188
				SYNTH	1189
				SYNTH	1190
				SYNTH	1191
				SYNTH	1192
				SYNTH	1193
				SYNTH	1194
				SYNTH	1195
				SYNTH	1196
				SYNTH	1197
				SYNTH	1198

	40GY 1*24X/*C*15X*1 ONE RESEARCH PARK*18X*1*24X/*C*15X*1 PALO	SYNTH	1199
	55 VERDES PENINSULA. CA 90274 1*24X/*C*15X*1 TEL: (213) 377-481	SYNTH	1200
	61* EXT. 322*6X*1*24X/*C*15X*1*38X*1*24X/*C*15X,40(1H-),24X/*C*79X)	SYNTH	1201
C	470 FORMAT (/12X*GENERALIZED KINETICS SYNTHESIS CODE: DR. WILLIAM B. L	SYNTH	1202
	1ACINA, NORTHROP RESEARCH AND TECHNOLOGY. DATE:*All)	SYNTH	1203
C	500 FORMAT (*11X,5HNDOT(,1*,13*,*,9H) = NDOT(,1*,13*,*,1H),*,12*,*A4)*)	SYNTH	1204
C	502 FORMAT (11X*PHI(*,12*,*,*,12*,*) = PHI(*,12*,*,*,12*,*)*10A4,6X)	SYNTH	1205
C	503 FORMAT (6X,*R = *.62A1,8X/(5X,*S*10X,56A1,8X))	SYNTH	1206
C	504 FORMAT (6X*DIMENSION PHI(N,1), NO(1)*49X)	SYNTH	1207
C	505 FORMAT (*C*5X*THIS SUBROUTINE WAS SYNTHESIZED BY EDITING AN INPUT	SYNTH	1208
	IFILE OF SYM-*10X/*C*5X*BOLOC REACTIONS WHICH DEFINE A COUPLED SYST	SYNTH	1209
	2EM OF ELECTRON AND*13X/*C*5X*MOLECULAR KINETICS EQUATIONS. IT RET	SYNTH	1210
	3URNS THE JACOBIAN MATRIX,*11X/*C*79X/*C*20X,*PHI(I,J) = D(NDOT(I)	SYNTH	1211
	4/D(INO(J))*29X/*C*79X/*C*5X*WHERE I,J = 1,2,3,...NTYPE. N IS THE D	SYNTH	1212
	5IMENSION DECLARATOR FOR PHI*8X/*C*5X*IN THE CALLING PROGRAM. THE	SYNTH	1213
	6RATE CONSTANTS KF AND KR HAVE UNITS*10X/*C*5X*OF CM2, SEC-1, CM3/S	SYNTH	1214
	7EC, CM6/SEC. ... AS APPROPRIATE,*,22X)	SYNTH	1215
C	506 FORMAT (*C*5X*EXTERNAL CIRCUIT EQUATIONS (Q * NO(*,12,*), AND I *	SYNTH	1216
	1NO(*,12,*)) -- *15X)	SYNTH	1217
C	507 FORMAT (6X*DO 1 I = 1,N*,62X/6X*DO 1 J = 1,N*,62X/4X*1 PHI(I,J) =	SYNTH	1218
	10*61X/6X*PHI(1,1) = - C*GAMMA*54X)	SYNTH	1219
C	508 FORMAT (6X*NE = NO(2)*64X/6X*IF (NE.EQ.0.) NE = 1.0*52X/6X*CONDUCT	SYNTH	1220
	1 = NE*E0*MU*56X/6X*RD = D/AREA/CONDUCT*55X/*C*79X/6X*IF (L0.EQ.0.)	SYNTH	1221
	2 GO TO 2*53X)	SYNTH	1222
C	509 FURMAT (6X*DQDT = NDOT(*12*) = NO(*12*)*)	SYNTH	1223
C	510 FORMAT (*DECK,JACOB#69X/6X*SUBROUTINE JACOB (N, T, NO, PHI)*42X)	SYNTH	1224
C	511 FORMAT (6X*PHI(*,12*,*,12*) = 0.*59X)	SYNTH	1225
C	512 FORMAT (6X*COMMON / CONST / NTOY, TMOL, FREQ, HNU*36X/6X*COMMON /	SYNTH	1226
	1DISCH / L0, C0, R0, MU, AREA, D*34X/6X*COMMON / SOURCE / UPLUS, JB	SYNTH	1227
	2EAM, DVDX, DEPOSIT, ENERGY, SB, S0*13X/6X*COMMON / GAINS / ALPHA,	SYNTH	1228
	3GAMMA, GAIN, ABSORB, OMEGA, LENGTH, CAVITY*8X/*C*79X)	SYNTH	1229
C	513 FORMAT (6X*DIOT = NDOT(*12*) = (-NO(*12*)/C0 - (R0 + RD)*1H*,*NO(*	SYNTH	1230
	112*),/L0*22X)	SYNTH	1231
C	514 FORMAT (4X*2 DQDT = NDOT(*12*) = - NO(*12*)/C0/(R0 + RD)*)	SYNTH	1232
C	515 FORMAT (6X*CURRENT = NO(*12*) = NDOT(*12*)*)	SYNTH	1233
C	516 FURMAT (6X*PHI(*12*,*12*) = 1.0*)	SYNTH	1234
C	517 FORMAT (6X*PHI(*12*,*12*) = -1./L0/C0*52X)	SYNTH	1235
C	518 FORMAT (6X*PHI(*12*,*12*) = -(R0 + RD)/L0*48X)	SYNTH	1236



C	519 FORMAT (6X*PHI(*I2*, 2) = NO(*I2*)*IH*,*RD/NE/L0*46X)	SYNTH	1256
C	520 FORMAT (6X*PHI(*I2*, 2) = NO(*I2*)*DIDQ*RD/NE/(R0 + RD)*)	SYNTH	1257
C	521 FORMAT (4X*2 PHI(*I2*,*I2*) = DIDQ = - 1./C0/(R0 + RD)*)	SYNTH	1258
C	600 FORMAT (*DECK,LEVELS#68X/6X*SUBROUTINE LEVELS (N1, N2, NO)*44X)	SYNTH	1259
C	601 FORMAT (6X,*REAL N1(1), N2(1), NO(1)*)	SYNTH	1260
C	602 FORMAT (6X,*N*,I1,*(*I2,*) = NO(*I2,*)*)	SYNTH	1261
C	603 FORMAT (6X,*N*,I1,*(*I2,*) = 0.*)	SYNTH	1262
C	604 FORMAT (*C THIS SUBROUTINE DETERMINES THE POPULATION DENSITIES	SYNTH	1263
	1 N1(I), N2(I)*10X/*C*5X*OF THE (LOWER AND UPPER) LEVELS INVOLVED I	SYNTH	1264
	2N THE ITH INELASTIC*13X/*C*5X*SCATTERING PROCESS INCLUDED IN THE C	SYNTH	1265
	2OUPLED E- KINETICS ANALYSIS.*10X)	SYNTH	1266
C	605 FORMAT (*C*5X*THIS SUBROUTINE WAS SYNTHESIZED BY EDITING AN INPUT	SYNTH	1267
	1FILE OF SYM-*10X/*C*5X*BOLOC REACTIONS WHICH DEFINE A COUPLED SYST	SYNTH	1268
	2EM OF ELECTRON AND*13X/*C*5X*MOLECULAR KINETICS. IT RETURNS THE RA	SYNTH	1269
	3TES NDOT(I) = (D/DT)NO(I),*11X/*C*5X*I = 1,2,...NTYPE (CM-3/SEC).	SYNTH	1270
	4RATE CONSTANTS KF AND KR HAVE UNITS*10X/*C*5X*OF CM2, SEC-1, CM3/S	SYNTH	1271
	5EC, CM6/SEC, ... AS APPROPRIATE.*22X)	SYNTH	1272
C	-----	SYNTH	1273
C	RETURN	SYNTH	1274
C	END	SYNTH	1275
C		SYNTH	1276
C		SYNTH	1277
C		SYNTH	1278
C		SYNTH	1279
C		SYNTH	1280
C		SYNTH	1281
C		SYNTH	1282
C		SYNTH	1283
C		SYNTH	1284
C		SYNTH	1285
C		SYNTH	1286

	SUBROUTINE ANALYZE (NTYPE, KTYPE, RATE, NTIME, RPCT, FLAG, PMAX,	ANALYZE	2
	1 GAS, PCT, KAPTION, LTAPE, MTAPE, NTAPE)	ANALYZE	3
C	.....	ANALYZE	4
C	.....	ANALYZE	5
C	.....	ANALYZE	6
C	THIS SUBROUTINE ANALYZES THE CONTRIBUTIONS OF ALL REACTIONS TO	ANALYZE	7
C	EVERY SPECIES, AND PRINTS OUT DIAGNOSTICS SUMMARIZING THE SENSI-	ANALYZE	8
C	TIVITY OF EACH REACTION TO THE TOTAL CALCULATION.	ANALYZE	9
C	.....	ANALYZE	10
C	.....	ANALYZE	11
C	.....	ANALYZE	12
C	DIMENSION RATE(1), NTIME(1), KAPTION(4), KODE(10), PMAX(1),	ANALYZE	13
	1 RPCT(1), FLAG(1), GAS(1)	ANALYZE	14
C	LOGICAL FLAG, TEST	ANALYZE	15
C	REWIND LTAPE	ANALYZE	16
	REWIND MTAPE	ANALYZE	17
	REWIND NTAPE	ANALYZE	18
	DO 8 I = 1,NTYPE	ANALYZE	19
	8 PMAX(I) = 0.	ANALYZE	20
	DO 1 K = 1,KTYPE	ANALYZE	21
	R = RATE(K)	ANALYZE	22
	READ (MTAPE) (NTIME(L), L = 1,NTYPE)	ANALYZE	23
	DO 2 I = 1,NTYPE	ANALYZE	24
	RPCT(I) = 0.	ANALYZE	25
	IF (R.EQ.0.) GO TO 2	ANALYZE	26
	NI = NTIME(I)	ANALYZE	27
	IF (NI.EQ.0) GO TO 2	ANALYZE	28
	RPCT(I) = NI*R	ANALYZE	29
	PABS = ABS(RPCT(I))	ANALYZE	30
	IF (PABS.GT.PMAX(I)) PMAX(I) = PABS	ANALYZE	31
	2 CONTINUE	ANALYZE	32
	1 WRITE (MTAPE) (RPCT(L), L = 1,NTYPE)	ANALYZE	33
	REWIND NTAPE	ANALYZE	34
		ANALYZE	35
		ANALYZE	36



C	DO 3 K = 1, KTYPE	ANALYZE	37
	PABS = 0.	ANALYZE	38
	READ (LTAPE) (RPCT(L), L = 1, NTYPE)	ANALYZE	39
	DO 4 I = 1, NTYPE	ANALYZE	40
	IF (PMAX(I).EQ.0.) GO TO 4	ANALYZE	41
	PERCENT = RPCT(I) * 100. / PMAX(I)	ANALYZE	42
	PERCENT = ABS(PERCENT)	ANALYZE	43
	IF (PERCENT.GT.PCT) FLAG(K) = .FALSE.	ANALYZE	44
	IF (PERCENT.GT.PABS) PABS = PERCENT	ANALYZE	45
	4 CONTINUE	ANALYZE	46
	3 WRITE (LTAPE) (RPCT(L), L = 1, NTYPE), PABS	ANALYZE	47
C	NA = 1	ANALYZE	48
	5 NB = NA + 9	ANALYZE	49
	IF (NB.GT.NTYPE) NB = NTYPE	ANALYZE	50
	NUASH = 37 + 11 * (NB - NA)	ANALYZE	51
	NX = (138 - NDASH) / 2	ANALYZE	52
	REWIND LTAPE	ANALYZE	53
	TEST = .FALSE.	ANALYZE	54
	DO 6 K = 1, KTYPE	ANALYZE	55
	K1 = K - 1	ANALYZE	56
	IF (K1.NE.50 * (K1 / 50)) GO TO 9	ANALYZE	57
	IF (K1.EQ.0) GO TO 7	ANALYZE	58
	WRITE (6, 100) NX, NDASH	ANALYZE	59
	IF (TEST) WRITE (6, 101) PCT	ANALYZE	60
	7 WRITE (6, 102) KAPTION, NX, (GAS(I), I = NA, NB)	ANALYZE	61
	WRITE (6, 100) NX, NDASH	ANALYZE	62
	TEST = .FALSE.	ANALYZE	63
	9 READ (LTAPE) (RPCT(L), L = 1, NTYPE), PCTMAX	ANALYZE	64
	R = RATE(K)	ANALYZE	65
	NFLAG = 1H	ANALYZE	66
	IF (FLAG(K)) NFLAG = 1H	ANALYZE	67
	IF (FLAG(K)) TEST = .TRUE.	ANALYZE	68
	DO 10 I = NA, NB	ANALYZE	69
	I1 = I - NA + 1	ANALYZE	70
	KODE(I1) = 1H	ANALYZE	71
	IF (RPCT(I1).EQ.0.) GO TO 10	ANALYZE	72
	ENCODE (10, 104, KODE(I1)) RPCT(I1)	ANALYZE	73
	10 CONTINUE	ANALYZE	74
	6 WRITE (6, 103) NX, NFLAG, K, R, PCTMAX, (KODE(I), I = 1, I1)	ANALYZE	75
	WRITE (6, 100) NX, NDASH	ANALYZE	76
	IF (TEST) WRITE (6, 101) PCT	ANALYZE	77
	NA = NB + 1	ANALYZE	78
	IF (NA.LE.NTYPE) GO TO 5	ANALYZE	79
C	-----	ANALYZE	80
C	-----	ANALYZE	81
C	-----	ANALYZE	82
	100 FORMAT (/ = X, = (1H - 1) /)	ANALYZE	83
C	-----	ANALYZE	84
	101 FORMAT (18X = * THIS REACTION CONTRIBUTES LESS THAN #F3.0% TO ALL	ANALYZE	85
	15 SPECIES THROUGHOUT THE ENTIRE CALCULATION SO FAR*)	ANALYZE	86
C	-----	ANALYZE	87
	102 FORMAT (1H1, 47X, 4A10, / 29X * PERCENTAGE CONTRIBUTION OF REACTION K TO	ANALYZE	88
	1 DN(I) / DT, EXPRESSED (FOR EACH SPECIES) * / 33X * AS A PERCENTAGE OF TH	ANALYZE	89
	2E MAXIMUM RATE OCCURRING FOR ALL REACTIONS INCLUDED * / = X, 4X * K * 4X,	ANALYZE	90
	3 * RATE(K) * 4X * MAX % * 5X, 8(1X, A10), A10, A7)	ANALYZE	91
C	-----	ANALYZE	92
	103 FORMAT (= X, A1, 14, 1PE12.3, 0PF8.1, 10(1X, A10))	ANALYZE	93
C	-----	ANALYZE	94
	104 FORMAT (F10, 3)	ANALYZE	95
C	-----	ANALYZE	96
C	-----	ANALYZE	97
C	-----	ANALYZE	98
	RETURN	ANALYZE	99
	END	ANALYZE	100
		ANALYZE	101
		ANALYZE	102

	SUBROUTINE DECODE (NAME, IMAGE, LHS, RHS, LABEL, GAS, NSIZE,	DECODE	2
	1 NTYPE, LONG)	DECODE	3
C	DIMENSION NAME(1), IMAGE(1), GAS(5,2), LABEL(5,2), KAR(10)	DECODE	4
	INTEGER LHS, RHS, GAS, E, HNU	DECODE	5
	E = 4HE(-)	DECODE	6
	HNU = 3MHNU	DECODE	7
	N0 = NTYPE	DECODE	8
	DO 1 L = 1,5	DECODE	9
	DO 1 M = 1,2	DECODE	10
	GAS(L,M) = 1H	DECODE	11
	1 LABEL(L,M) = 0	DECODE	12
	M = 1	DECODE	13
	I = J = N = MM = 0	DECODE	14
	2 IF (I.EQ.LONG) GO TO 4	DECODE	15
	I = I+1	DECODE	16
	IF (IMAGE(I).EQ.1H) GO TO 2	DECODE	17
	IF (IMAGE(I).NE.1H) GO TO 3	DECODE	18
	MM = 1	DECODE	19
	GO TO 4	DECODE	20
	3 IF (IMAGE(I).NE.1H) GO TO 6	DECODE	21
	IF (IMAGE(I+1).EQ.1H).OR.IMAGE(I+1).EQ.1H) GO TO 6	DECODE	22
	MM = 0	DECODE	23
	GO TO 4	DECODE	24
	6 IF (J.EQ.NSIZE) GO TO 2	DECODE	25
	J = J+1	DECODE	26
	KAR(J) = IMAGE(I)	DECODE	27
	GO TO 2	DECODE	28
	4 IF (J.EQ.0) GO TO 99	DECODE	29
	IF (M.GT.2) GO TO 99	DECODE	30
	ENCODE (10,100,NGAS) (KAR(L), L = 1,J)	DECODE	31
	100 FORMAT (10A1)	DECODE	32
	IF (NGAS.EQ.1HE.OR.NGAS.EQ.2HE-1) NGAS = E	DECODE	33
	J = 0	DECODE	34
	N = N+1	DECODE	35
	GAS(N,M) = NGAS	DECODE	36
	IF (NGAS.EQ.HNU) GO TO 7	DECODE	37
	IF (N0.EQ.0) GO TO 9	DECODE	38
	DO 5 L = 1,N0	DECODE	39
	IF (NGAS.NE.NAME(L)) GO TO 5	DECODE	40
	LABEL(N,M) = L	DECODE	41
	GO TO 7	DECODE	42
	5 CONTINUE	DECODE	43
	9 N0 = N0+1	DECODE	44
	NAME(N0) = NGAS	DECODE	45
	LABEL(N,M) = N0	DECODE	46
	7 M = N+MM	DECODE	47
	IF (MM.EQ.1) N = 0	DECODE	48
	GO TO 2	DECODE	49
	99 NTYPE = N0	DECODE	50
C		DECODE	51
C	GENERATE CHECKSUM IDENTIFIERS --	DECODE	52
C		DECODE	53
	K1 = K2 = K1SQ = K2SQ = 0	DECODE	54
	DO 8 L = 1,5	DECODE	55
	K1 = K1 + LABEL(L,1)	DECODE	56
	K2 = K2 + LABEL(L,2)	DECODE	57
	K1SQ = K1SQ + LABEL(L,1)*LABEL(L,1)	DECODE	58
	8 K2SQ = K2SQ + LABEL(L,2)*LABEL(L,2)	DECODE	59
	ENCODE (10,110,LHS) K1, K1SQ	DECODE	60
	ENCODE (10,110,RHS) K2, K2SQ	DECODE	61
	110 FORMAT (I4,I6)	DECODE	62
C		DECODE	63
	RETURN	DECODE	64
	END	DECODE	65
		DECODE	66

	SUBROUTINE UPDATE (INFILE, NTAPE, NSCRTCH, LIST, DATE)	UPDATE	2
C		UPDATE	3
C	.....	UPDATE	4
C		UPDATE	5
C	THIS SUBROUTINE SEARCHES TWO SOURCES--AN INPUT FILE TAPE *INFILE*	UPDATE	6
C	AND/OR INPUT CARD DATA (IF MODIFY = TRUE)--TO GENERATE AN UPDATED	UPDATE	7
C	FILE ON TAPE *NTAPE*, WHICH CONTAINS ALL OF THE DATA OF THE FILE	UPDATE	8
C	*INFILE* MODIFIED WITH ADDITIONS OR REVISIONS DEFINED BY THE CARD	UPDATE	9
C	DATA. THE FILE ON TAPE *NTAPE* CAN BE CATALOGUED AS A PERMANENT	UPDATE	10
C	FILE, IF DESIRED, FOR FUTURE USE AS THE INPUT LIBRARY. THE FILE	UPDATE	11
C	GENERATED ON NTAPE CONTAINS DATES OF ENTRY FOR ALL CROSS SECTIONS	UPDATE	12
C	WHICH HAVE BEEN CATALOGUED. IF LIST = TRUE, THE CONTENTS OF THE	UPDATE	13
C	UPDATED CROSS SECTION FILE *NTAPE* ARE PRINTED OUT.	UPDATE	14
C		UPDATE	15
C	.....	UPDATE	16
C		UPDATE	17
	DIMENSION IMAGE(8), KINETIC(60), NAME(100), LABEL(5,2), GAS(5,2)	UPDATE	18
	INTEGER BLANK, LHS1, RHS1, LHS2, RHS2, DATE, GAS	UPDATE	19
	LOGICAL LIST, MODIFY, ENDFILE	UPDATE	20
C		UPDATE	21
	CALL SECOND (T0)	UPDATE	22
	NTYPE = 0	UPDATE	23
	NUMRERS = 10H1234567890	UPDATE	24
	BLANK = 1H	UPDATE	25
	REWIND NTAPE	UPDATE	26
	ENDFILE = .FALSE.	UPDATE	27
	INPUT = INFILE	UPDATE	28
C		UPDATE	29
C	GENERATE OR MODIFY ELECTRON CROSS SECTION DATA FILE --	UPDATE	30
C		UPDATE	31
	MODIFY = .TRUE.	UPDATE	32
	READ (5,100)	UPDATE	33
	IF (EOF(5)) 10,20	UPDATE	34
10	MODIFY = .FALSE.	UPDATE	35
20	REWIND 5	UPDATE	36
	REWIND INPUT	UPDATE	37
	READ (INPUT,100)	UPDATE	38
	IF (EOF(INPUT)) 1,2	UPDATE	39
	1 INPUT = 0	UPDATE	40
	GO TO 3	UPDATE	41
	2 BACKSPACE INPUT	UPDATE	42
	3 IF (MODIFY.AND.INPUT.EQ.INFILE) GO TO 33	UPDATE	43
	NFILE = INPUT	UPDATE	44
	IF (INPUT.EQ.INFILE) GO TO 46	UPDATE	45
	NFILE = 5	UPDATE	46
	IF (.NOT.MODIFY) RETURN	UPDATE	47
C		UPDATE	48
C	THE FOLLOWING SECTION IS USED WITH ONLY ONE INPUT DATA SOURCE --	UPDATE	49
C		UPDATE	50
	46 READ (NFILE,120) IMAGE	UPDATE	51
	IF (EOF(NFILE)) 45,4	UPDATE	52
	4 IF (IMAGE(8).EQ.BLANK) IMAGE(8) = DATE	UPDATE	53
	WRITE (NTAPE,120) IMAGE	UPDATE	54
	READ (NFILE,120) IMAGE	UPDATE	55
	WRITE (NTAPE,120) IMAGE	UPDATE	56
	NREC = 0	UPDATE	57
	47 READ (NFILE,120) IMAGE	UPDATE	58



WRITE (NTAPE,120) IMAGE	UPDATE	59
IF (IMAGE(1).NE.BLANK) GO TO 14	UPDATE	60
IF (NREC.GT.0) GO TO 46	UPDATE	61
BACKSPACE NTAPE	UPDATE	62
BACKSPACE NTAPE	UPDATE	63
BACKSPACE NTAPE	UPDATE	64
GO TO 46	UPDATE	65
14 NREC = 1	UPDATE	66
READ (NFILE,120) IMAGE	UPDATE	67
WRITE (NTAPE,120) IMAGE	UPDATE	68
GO TO 47	UPDATE	69
C THE FOLLOWING SECTION OCCURS WHEN DATA IS ASSEMBLED FROM BOTH A	UPDATE	70
C TAPE AND CARD INPUT FILE --	UPDATE	71
C	UPDATE	72
33 IF (ENDFILE) GO TO 44	UPDATE	73
READ (INPUT,150) KINETIC	UPDATE	74
IF (EOF(INPUT)) 6,7	UPDATE	75
7 CALL DECODE (NAME, KINETIC, LHS1, RHS1, LABEL, GAS, 10, NTYPE, 60)	UPDATE	76
GO TO 37	UPDATE	77
6 ENDFILE = .TRUE.	UPDATE	78
INPUT = 5	UPDATE	79
NFILE = 5	UPDATE	80
REWIND NFILE	UPDATE	81
44 READ (NFILE,120) IMAGE	UPDATE	82
IF (EOF(NFILE)) 45,8	UPDATE	83
8 IMAGE(8) = DATE	UPDATE	84
BACKSPACE NFILE	UPDATE	85
READ (NFILE,150) KINETIC	UPDATE	86
CALL DECODE (NAME, KINETIC, LHS1, RHS1, LABEL, GAS, 10, NTYPE, 60)	UPDATE	87
C CHECK TAPE3 TO DETERMINE WHETHER THE PROCESS ENCOUNTERED ON CARD	UPDATE	88
C INPUT WAS PREVIOUSLY USED TO UPDATE TAPE FILE DATA --	UPDATE	89
C	UPDATE	90
REWIND NSCRTCH	UPDATE	91
43 READ (NSCRTCH) LHS2, RHS2	UPDATE	92
IF (EOF(NSCRTCH)) 34,5	UPDATE	93
5 IF (LHS2.NE.LHS1.OR.RHS2.NE.RHS1) GO TO 43	UPDATE	94
READ (INPUT,120)	UPDATE	95
GO TO 41	UPDATE	96
37 NFILE = INPUT	UPDATE	97
BACKSPACE NFILE	UPDATE	98
READ (NFILE,120) IMAGE	UPDATE	99
C CHECK CARDS TO SEE IF A CROSS SECTION PRESENTED ON THE TAPE FILE	UPDATE	100
C SHOULD BE SUPERCEDED BY CARD INPUT DATA (UPDATE) --	UPDATE	101
C	UPDATE	102
REWIND 5	UPDATE	103
36 READ (5,150) KINETIC	UPDATE	104
IF (EOF(5)) 34,9	UPDATE	105
9 CALL DECODE (NAME, KINETIC, LHS2, RHS2, LABEL, GAS, 10, NTYPE, 60)	UPDATE	106
IF (LHS2.NE.LHS1.OR.RHS2.NE.RHS1) GO TO 35	UPDATE	107
C THE PROCESS DEFINED ON TAPE FILE HAS BEEN FOUND IN THE CARD INPUT.	UPDATE	108
C SO IT IS REPLACED --	UPDATE	109
C	UPDATE	110
NFILE = 5	UPDATE	111
	UPDATE	112
	UPDATE	113
	UPDATE	114
	UPDATE	115

BACKSPACE 5	UPDATE	116
READ (5,120) IMAGE	UPDATE	117
IMAGE(R) = OATE	UPDATE	118
WRITE (NSCRTCH) LHS2, RHS2	UPDATE	119
READ (INPUT,120)	UPDATE	120
GO TO 34	UPDATE	121
C	UPDATE	122
35 READ (5,120)	UPDATE	123
31 READ (5,120) KARD	UPDATE	124
IF (KARD.EQ.BLANK) GO TO 36	UPDATE	125
READ (5,120)	UPDATE	126
GO TO 31	UPDATE	127
C	UPDATE	128
C	UPDATE	129
C COPY CROSS SECTION DATA ONTO TAPE NTAPE --	UPDATE	130
34 WRITE (NTAPE,120) IMAGE	UPDATE	131
READ (NFILE,120) IMAGE	UPDATE	132
WRITE (NTAPE,120) IMAGE	UPDATE	133
NREC = 0	UPDATE	134
32 READ (NFILE,120) IMAGE	UPDATE	135
WRITE (NTAPE,120) IMAGE	UPDATE	136
IF (IMAGE(1).NE.BLANK) GO TO 13	UPDATE	137
IF (NREC.GT.0) GO TO 39	UPDATE	138
BACKSPACE NTAPE	UPDATE	139
BACKSPACE NTAPE	UPDATE	140
BACKSPACE NTAPE	UPDATE	141
GO TO 39	UPDATE	142
13 NREC = 1	UPDATE	143
READ (NFILE,120) IMAGE	UPDATE	144
WRITE (NTAPE,120) IMAGE	UPDATE	145
GO TO 32	UPDATE	146
39 IF (NFILE.EQ.INPUT) GO TO 33	UPDATE	147
C	UPDATE	148
C EXHAUST OLD DATA FOR THIS PROCESS --	UPDATE	149
C	UPDATE	150
41 READ (INPUT,120) IMAGE	UPDATE	151
IF (IMAGE(1).EQ.BLANK) GO TO 33	UPDATE	152
READ (INPUT,120)	UPDATE	153
GO TO 41	UPDATE	154
C	UPDATE	155
C IF ELECTRON CROSS SECTION DATA CONTAINED MODIFICATIONS BY CARD	UPDATE	156
C INPUT, A NEW FILE IS GENERATED. THE CONTENTS OF THE UPDATED FILE	UPDATE	157
C (WHICH MAY BE CATALOGUED FOR FUTURE USE) ARE COPIED ONTO OUTPUT	UPDATE	158
C IF LIST IS SPECIFIED TO BE TRUE.	UPDATE	159
C	UPDATE	160
45 ENDFILE NTAPE	UPDATE	161
REWIND NSCRTCH	UPDATE	162
CALL SECOND (TIME)	UPDATE	163
TIME = TIME-T0	UPDATE	164
IF (MODIFY) WRITE (6,160) TIME	UPDATE	165
IF (.NOT.LIST) GO TO 99	UPDATE	166
IF (NFILE.EQ.NFILE) GO TO 99	UPDATE	167
REWIND NTAPE	UPDATE	168
LL = LINE = 0	UPDATE	169
23 READ (NTAPE,120) IMAGE	UPDATE	170
IF (EOF(NTAPE)) 11,12	UPDATE	171
12 IF (LINE.NE.0) GO TO 25	UPDATE	172





	SUBROUTINE PLASMA (NDATA, MAX, MESH, LHS, RHS, PROCESS, EV, F,	PLASMA	2
	1 G, Q, U0, UM, NTYPE, NAME, MISSING, ERROR, OUTSIDE, IDEG, OUT)	PLASMA	3
C	-----	PLASMA	4
C		PLASMA	5
C		PLASMA	6
C	THIS SUBROUTINE SCANS THE ELECTRON CROSS SECTION FILE TO EXTRACT	PLASMA	7
C	DATA FOR THE INPUT REACTION *PROCESS*, DEFINED BY *LHS* AND *RHS*.	PLASMA	8
C	IF THE REACTION IS FOUND, THE RAW CROSS SECTION DATA IS EXAMINED	PLASMA	9
C	FOR ERRORS, AND IF ACCEPTABLE, IS INTERPOLATED OVER THE INPUT EN-	PLASMA	10
C	ERGY GRID DEFINED BY THE VECTOR EV(I), I = 1,MESH+1. THE EXTERNAL	PLASMA	11
C	ELECTRON FILE CONSISTS OF (ARBITRARILY MANY) PACKAGES OF THE FORM	PLASMA	12
C		PLASMA	13
C	A) REACTION, UNITS, NPTS, MONTH (6A1,F7,3,I3,A10)	PLASMA	14
C		PLASMA	15
C	B) COMMENT (6A10)	PLASMA	16
C		PLASMA	17
C	1) ENERGY VALUES (EV)	PLASMA	18
C	2) CROSS SECTION VALUES	PLASMA	19
C		PLASMA	20
C	ARBITRARY NUMBER OF CARD PAIRS (1) AND (2), TERMINATED	PLASMA	21
C	BY THE BLANK CARD (C) BELOW. THE FORMAT IS VARIABLE	PLASMA	22
C	THERE ARE *NPTS* FIELDS FK,0, WHERE K = 180/NPTS).	PLASMA	23
C	IF NPTS ≤ 0 OR NPTS > 10, PROGRAM DEFAULTS TO NPTS = 10.	PLASMA	24
C	THE UNITS OF THE CROSS SECTION DATA ARE UNITS X 1.0E-16	PLASMA	25
C	CM2 (DEFAULT: UNITS = 1).	PLASMA	26
C		PLASMA	27
C	C) BLANK CARD	PLASMA	28
C		PLASMA	29
C	ERROR CONDITIONS ENCOUNTERED ARE IDENTIFIED BY LOGICAL VARIABLES	PLASMA	30
C	WHICH ARE RETURNED WITH THE VALUE .TRUE. TO THE CALLING PROGRAM.	PLASMA	31
C		PLASMA	32
C	INPUT PARAMETERS --	PLASMA	33
C		PLASMA	34
C	NDATA = LOGICAL FILE FOR ELECTRON CROSS SECTION DATA.	PLASMA	35
C		PLASMA	36
C	MAX = DIMENSION DECLARATOR DEFINED FOR EV(I), F(I), G(I),	PLASMA	37
C	AND Q(I) IN THE CALLING PROGRAM.	PLASMA	38
C		PLASMA	39
C	MESH = NUMBER OF SUBINTERVALS INTO WHICH THE ELECTRON ENERGY	PLASMA	40
C	RANGE IS DIVIDED (AND OVER WHICH THE CROSS SECTION DATA	PLASMA	41
C	IS TO BE INTERPOLATED). MESH+1 ≤ MAX.	PLASMA	42
C		PLASMA	43
C	PROCESS = VECTOR (4A10) CONTAINING MOLLERITH NAME OF REACTION.	PLASMA	44
C		PLASMA	45
C	LHS,RHS = (INTEGERS) ENCODED WITH UNIQUE IDENTIFIERS OF THE LEFT	PLASMA	46
C	AND RIGHT HAND SIDE OF THE REACTION (CF. SUBROUTINE	PLASMA	47
C	DECODE).	PLASMA	48
C		PLASMA	49
C	F, G = SCRATCH VECTORS (DIMENSIONED MAX IN CALLING PROGRAM.)	PLASMA	50
C		PLASMA	51
C		PLASMA	52
C	OUTPUT PARAMETERS --	PLASMA	53
C		PLASMA	54
C	Q(I) = CROSS SECTION VECTOR (UNITS OF CM2) DEFINED AT EV(I)	PLASMA	55
C		PLASMA	56
C	U0 = CROSS SECTION THRESHOLD ENERGY (EV).	PLASMA	57
C		PLASMA	58

C		PLASMA	59
C	UM = CROSS SECTION MAXIMUM ENERGY CUTOFF (EV).	PLASMA	60
C		PLASMA	61
C	MISSING = .TRUE., IF THE REACTION WAS NOT FOUND.	PLASMA	62
C		PLASMA	63
C	OUTSIDE = .TRUE., IF THE CROSS SECTIONS WERE DEFINED OVER AN	PLASMA	64
C	ENERGY RANGE (U0,UM) THAT DOES NOT SPAN THE ENERGY GRID	PLASMA	65
C	DEFINED BY THE INPUT ENERGY VECTOR EV(I).	PLASMA	66
C		PLASMA	67
C	ERROR = .TRUE., IF ENERGY VALUES WERE NOT SEQUENCED IN MONO-	PLASMA	68
C	TONICALLY ASCENDING ORDER (ONLY DATA IN ASCENDING ORDER	PLASMA	69
C	IS PERMITTED.)	PLASMA	70
C		PLASMA	71
C	-----	PLASMA	72
C		PLASMA	73
	DIMENSION EV(1), F(1), G(1), Q(1), NAME(1), IMAGE(60), KOMMENT(6),	PLASMA	74
	1 PROCESS(4), Y(10), LABEL(5,2), GAS(5,2)	PLASMA	75
C		PLASMA	76
	LOGICAL FORWARD, REVERSE, ERROR, THRESH, OUT(2), MISSING, OUTSIDE	PLASMA	77
	INTEGER LHS, RHS, GAS	PLASMA	78
C		PLASMA	79
	UNIT = 1.E-16	PLASMA	80
	NPAGE = 5H(1H1)	PLASMA	81
	ERROR = OUTSIDE = .FALSE.	PLASMA	82
	MESHP1 = MESH+1	PLASMA	83
	DO 17 L = 1,MESHP1	PLASMA	84
	17 Q(L) = 0.	PLASMA	85
C		PLASMA	86
	REWIND NDATA	PLASMA	87
	20 READ (NDATA,100) IMAGE, UNITS, NPTS, MONTH	PLASMA	88
	100 FORMAT (60A1,F7.3,I3,A10)	PLASMA	89
	IF (EOF(NDATA)) 99,1	PLASMA	90
	1 CALL DECODE (NAME, IMAGE, L1, L2, LABEL, GAS, 10, NTYPE, 60)	PLASMA	91
	FORWARD = LHS.EQ.L1.AND.RHS.EQ.L2	PLASMA	92
	REVERSE = LHS.EQ.L2.AND.RHS.EQ.L1	PLASMA	93
	MISSING = .NOT.(FORWARD.OR.REVERSE)	PLASMA	94
	IF ((NPTS.LE.0).OR.(NPTS.GT.10)) NPTS = 10	PLASMA	95
	INV = 80/NPTS	PLASMA	96
	ENCODE (10,101,FORM) NPTS, INV	PLASMA	97
	101 FORMAT (1H(,I2,1HE,I2,3H.0))	PLASMA	98
	IF (UNITS.EQ.0.) UNITS = 1.	PLASMA	99
	IF (MISSING) GO TO 3	PLASMA	100
	THRESH = .FALSE.	PLASMA	101
C	THRESH = FALSE AT START OF CROSS SECTION DATA FOR PROCESS J, AND	PLASMA	102
C	BECOMES TRUE AS SOON AS THE FIRST NON-ZERO VALUE APPEARS.	PLASMA	103
	DO 4 L = 1,MAX	PLASMA	104
	4 F(L) = G(L) = 0.	PLASMA	105
	LPTS = LTH = 1	PLASMA	106
	F0 = U0 = 0.	PLASMA	107
	X0 = - 1.0	PLASMA	108
	LAST = MAX	PLASMA	109
C		PLASMA	110
	3 READ (NDATA,112) KOMMENT	PLASMA	111
	112 FORMAT (8A10)	PLASMA	112
C		PLASMA	113
	ISW = 1	PLASMA	114
	L = 0	PLASMA	115

8 L2 = L	PLASMA	116
7 ISW = - ISW	PLASMA	117
READ (NDATA,FORM) (Y(N), N = 1,NPTS)	PLASMA	118
SUM = 0.	PLASMA	119
DO 16 N = 1,NPTS	PLASMA	120
16 SUM = SUM + Y(N)	PLASMA	121
C	PLASMA	122
C BLANK CARD TERMINATES (U, SIGMA) DATA PACKAGE FOR THE INELASTIC	PLASMA	123
C PROCESSES NK.	PLASMA	124
C	PLASMA	125
IF ((SUM.EQ.0.).AND.(ISW.LT.0)) GO TO 10	PLASMA	126
IF (MISSING) GO TO 7	PLASMA	127
IF (L2.EQ.LAST) GO TO 7	PLASMA	128
IF (ISW.GT.0) GO TO 9	PLASMA	129
C	PLASMA	130
L = L2	PLASMA	131
DO 12 N = 1,NPTS	PLASMA	132
IF (L.EQ.LAST) GO TO 7	PLASMA	133
IF (Y(N).GT.X0) GO TO 11	PLASMA	134
IF (Y(N).NE.0.) ERROR = .TRUE.	PLASMA	135
LAST = L	PLASMA	136
GO TO 7	PLASMA	137
11 L = L+1	PLASMA	138
12 X0 = F(L) = Y(N)	PLASMA	139
GO TO 7	PLASMA	140
C	PLASMA	141
9 L = L2	PLASMA	142
DO 13 N = 1,NPTS	PLASMA	143
IF (L.EQ.LAST) GO TO 8	PLASMA	144
L = L+1	PLASMA	145
G(L) = Y(N)*UNITS	PLASMA	146
IF (G(L).GT.0.) THRESH = .TRUE.	PLASMA	147
IF (THRESH) GO TO 5	PLASMA	148
U0 = F(L)	PLASMA	149
LTH = L	PLASMA	150
5 IF (F0.EQ.0.) GO TO 13	PLASMA	151
UM = F(L)	PLASMA	152
LPTS = L	PLASMA	153
13 F0 = G(L)	PLASMA	154
GO TO 8	PLASMA	155
C	PLASMA	156
10 IF (MISSING) GO TO 20	PLASMA	157
LPTS = LPTS - LTH + 1	PLASMA	158
ERROR = ERROR.OR.LPTS.LE.1	PLASMA	159
IF (ERROR) GO TO 6	PLASMA	160
DO 14 L = 1,LPTS	PLASMA	161
LL = L + LTH - 1	PLASMA	162
G(LL) = G(L)	PLASMA	163
14 F(LL) = F(LL)	PLASMA	164
C	PLASMA	165
OUTSIDE = .TRUE.	PLASMA	166
DO 15 L = 1,MESHP1	PLASMA	167
SIGMA = 0.	PLASMA	168
X = EV(L)	PLASMA	169
IF (X.GT.UM) GO TO 15	PLASMA	170
IF (X.LT.U0) GO TO 15	PLASMA	171
OUTSIDE = .FALSE.	PLASMA	172



CALL INTERP (IDEG, X, SIGMA, F, G, 1, LPTS)	PLASMA	173
15 IF (SIGMA.GE.0.) Q(L) = SIGMA*UNIT	PLASMA	174
C 6 IF (.NOT.OUT(1)) GO TO 18	PLASMA	175
WRITE (6,102) UNIT, PROCESS	PLASMA	176
102 FORMAT (1H1,34X,*CROSS SECTION (UNITS OF*, 1PE10.3* CM2) VS ELECTR	PLASMA	177
ION ENERGY (EV) FOR*/57X,4A10//)	PLASMA	178
IF (KOMMENT(1).NE.1H ) WRITE (6,103) KOMMENT	PLASMA	179
103 FORMAT (35X,*REFERENCE -- *,6A10//)	PLASMA	180
WRITE (6,104) UNIT	PLASMA	181
104 FORMAT (56X*U*15X*SIGMA(U)*54X*(EV) *10X*(1PE7.1* CM2)*)	PLASMA	182
WRITE (6,105)	PLASMA	183
105 FORMAT (/50X,33(1H-)/)	PLASMA	184
LL = 0	PLASMA	185
LINE = 1	PLASMA	186
DO 33 L = 1,LPTS	PLASMA	187
IF (LL.NE.LINE) GO TO 32	PLASMA	188
WRITE (6,NPAGE)	PLASMA	189
WRITE (6,104) UNIT	PLASMA	190
WRITE (6,105)	PLASMA	191
32 LINE = LINE+1	PLASMA	192
WRITE (6,106) F(L), G(L)	PLASMA	193
106 FORMAT (50X,F8.3,1PE22.3)	PLASMA	194
LL = 40*(LINE/40)	PLASMA	195
33 IF (LL.EQ.LINE) WRITE (6,105)	PLASMA	196
IF (LL.NE.LINE) WRITE (6,105)	PLASMA	197
WRITE (6,108) MONTH	PLASMA	198
108 FORMAT (/50X,*(DATA WAS SUBMITTED ON*,A9,*))*	PLASMA	199
C IF IOUT(2) IS SPECIFIED, DATA IS PLOTTED --	PLASMA	200
C	PLASMA	201
C	PLASMA	202
18 IF (.NOT.OUT(2)) GO TO 99	PLASMA	203
IF (ERROR) GO TO 99	PLASMA	204
IPLLOT = 0	PLASMA	205
IF (F(LPTS).GT.50.*F(1)) IPLLOT = -1	PLASMA	206
WRITE (6,102) UNIT, PROCESS	PLASMA	207
CALL PLOT (MAX, LPTS, 1, G, 0., 0., F, 0., 0., .TRUE., .TRUE.,	PLASMA	208
1 .TRUE., .TRUE., .TRUE., TITLE, 1, IPLLOT)	PLASMA	209
IF (IPLLOT.EQ.0) WRITE (6,109)	PLASMA	210
109 FORMAT (/64X,*ELECTRON ENERGY U (EV)*)	PLASMA	211
IF (IPLLOT.NE.0) WRITE (6,110)	PLASMA	212
110 FORMAT (/60X,*LOG OF ELECTRON ENERGY U (EV)*)	PLASMA	213
C 99 RETURN	PLASMA	214
END	PLASMA	215
	PLASMA	216
	PLASMA	217

FUNCTION SHAPE(T)	SHAPE	2
DIMENSION TIME(21), Y(21)	SHAPE	3
LOGICAL INTRP	SHAPE	4
COMMON / TIMES / TR, TF, TFALL, TDROP, TIME, Y, INTRP, N, UNITS	SHAPE	5
TP = T/UNITS	SHAPE	6
IF (INTRP) GO TO 2	SHAPE	7
SHAPE = 0.	SHAPE	8
IF (T.LT.0.) RETURN	SHAPE	9
SHAPE = 1.0	SHAPE	10
IF (TR.EQ.0.) RETURN	SHAPE	11
X = TR/TF	SHAPE	12
F0 = X*(1. + 1./X)**(1. + X)	SHAPE	13
IF (TP.GT.TDROP) GO TO 1	SHAPE	14
SHAPE = F0*(1. - EXP(-TP/TR))*EXP(-TP/TF)	SHAPE	15
RETURN	SHAPE	16
1 TP = TP-TDROP	SHAPE	17
F0 = F0*(1. - EXP(-TDROP/TR))*EXP(-TDROP/TF)	SHAPE	18
SHAPE = F0*EXP(-TP/TFALL)	SHAPE	19
RETURN	SHAPE	20
2 CALL INTERP (2, TP, SHAPE, TIME, Y, 1, N)	SHAPE	21
IF (SHAPE.LT.0.) SHAPE = 0.	SHAPE	22
RETURN	SHAPE	23
END	SHAPE	24

	SUBROUTINE BOLTZ (MAX, MESH, NK, GAS, FRACT, MIX, NMOL, TMOL,	BOLTZ	2
	1 ITMAX, TMAX, EPS, KAPTION, DATE, OUT, EVCM, NE, PROCESS, U, N1,	BOLTZ	3
	2 N2, NEL, S, SBEAM, SOURCE, X, XQ, QM, F, G, A, B, VSIG, POWER,	BOLTZ	4
	3 PCOLL, DISCH, DEPOSIT, DEDT, ELASTIC, DNEDT, DLNEDT, IONIZE,	BOLTZ	5
	4 ATTACH, VD, MU, D, EK, AMPS, UBAR, TE, CONVRGE, PBAL)	BOLTZ	6
C	-----	BOLTZ	7
C		BOLTZ	8
C		BOLTZ	9
C	THIS SUBROUTINE PERFORMS A NUMERICAL SOLUTION OF THE BOLTZMANN TRANS-	BOLTZ	10
C	PORT EQUATION FOR A MULTICOMPONENT GAS, WITH THE INCLUSION OF INELAS-	BOLTZ	11
C	TIC E-MOLECULE COLLISIONS, ELASTIC MOMENTUM TRANSFER COLLISIONS (WITH	BOLTZ	12
C	RECOIL), SUPERELASTIC COLLISIONS, ELECTRON-ELECTRON (COULOMB) SCAT-	BOLTZ	13
C	TERING, AND EXTERNAL ELECTRON ENERGY DEPOSITION. THE ANALYSIS CON-	BOLTZ	14
C	SISTS OF CALCULATION OF THE ELECTRON ENERGY DISTRIBUTION F(U) FOR THE	BOLTZ	15
C	ELECTRONS IN A PLASMA WITH IONIZATION NE/NTOT (OR SINGLE ELECTRON IF	BOLTZ	16
C	NE = 0), SUBJECTED TO A (SPATIALLY AND TEMPORALLY) CONSTANT ELECTRIC	BOLTZ	17
C	FIELD. THIS SUBROUTINE CALCULATES THE ELECTRON ENERGY DISTRIBUTION	BOLTZ	18
C	FUNCTION, PLASMA PARAMETERS (MU, D, VD, EK, UBAR, TE, ...), ELECTRON	BOLTZ	19
C	EXCITATION RATES (CM3/SEC) FOR THE FORWARD (AND REVERSE) INELASTIC	BOLTZ	20
C	COLLISION PROCESSES J = 1,2,...,NK, AND THE (NET) ELECTRICAL POWER	BOLTZ	21
C	PARTITIONING FOR ALL OF THESE PROCESSES (AND FOR ELASTIC HEATING).	BOLTZ	22
C		BOLTZ	23
C	INPUT PARAMETERS --	BOLTZ	24
C		BOLTZ	25
C	MAX = DIMENSION DECLARATOR DEFINED FOR VARIOUS VECTORS	BOLTZ	26
C	AND ARRAYS IN THE CALLING PROGRAM.	BOLTZ	27
C		BOLTZ	28
C	MESH = NUMBER OF SUBDIVISIONS INTO WHICH THE ELECTRON	BOLTZ	29
C	ENERGY RANGE (0,EMAX) IS PARTITIONED.	BOLTZ	30
C		BOLTZ	31
C	NK = NUMBER OF INELASTIC E(-) COLLISION PROCESSES	BOLTZ	32
C	INCLUDED IN THE PLASMA KINETICS ANALYSIS.	BOLTZ	33
C		BOLTZ	34
C	X(I) = (I - 1)DX, ELECTRON ENERGY GRID (EV), WHERE	BOLTZ	35
C	DX = EMAX/MESH.	BOLTZ	36
C		BOLTZ	37
C	XQ(I,J) = X(I)Q(I,J), WHERE Q(I,J) = INELASTIC SCATTERING	BOLTZ	38
C	CROSS SECTION (CM2) FOR THE JTH INELASTIC ELECTRON	BOLTZ	39
C	COLLISION PROCESS, AT ENERGY X(I).	BOLTZ	40
C		BOLTZ	41
C	QM(I,L) = ARRAY CONTAINING TWO COLUMN VECTORS, CONVENIENT	BOLTZ	42
C	FOR THE BOLTZMANN ANALYSIS --	BOLTZ	43
C		BOLTZ	44
C	QM(I,1) = X(I+.5)/NTOT SUMK(FI(K)*QMOM(I+.5,K))	BOLTZ	45
C		BOLTZ	46
C	QM(I,2) = X(I+.5)**2*2*ME*NTOT*	BOLTZ	47
C	SUMK(FI(K)*QMOM(I+.5,K)/MASS(K))	BOLTZ	48
C		BOLTZ	49
C	WHERE QMOM(I,K) = MOMENTUM TRANSFER CROSS SECTION	BOLTZ	50
C	AT ENERGY X(I) FOR SPECIES K.	BOLTZ	51
C		BOLTZ	52
C	F(I) = F(X(I)), INITIAL GUESS FOR THE ELECTRON ENERGY	BOLTZ	53
C	DISTRIBUTION FUNCTION, DIMENSIONED F(MAX) IN THE	BOLTZ	54
C	CALLING PROGRAM.	BOLTZ	55
C		BOLTZ	56
C	U(J) = ENERGY LOSS (EV) FOR THE JTH INELASTIC ELECTRON	BOLTZ	57
C	PROCESS.	BOLTZ	58

N1(J)	= NEUTRAL PARTICLE DENSITY (CM-3) OF THE LOWER STATE IN THE JTH INELASTIC COLLISION PROCESS.	BOLTZ	59
		BOLTZ	60
		BOLTZ	61
N2(J)	= NEUTRAL PARTICLE DENSITY (CM-3) OF THE UPPER STATE IN THE JTH INELASTIC COLLISION PROCESS.	BOLTZ	62
		BOLTZ	63
		BOLTZ	64
NEL(J)	= NET NUMBER OF ELECTRONS (RIGHT HAND SIDE - LEFT HAND SIDE) FOR THE JTH ELECTRON COLLISION PROCESS.	BOLTZ	65
		BOLTZ	66
		BOLTZ	67
NMOL	= TOTAL NEUTRAL PARTICLE DENSITY (CM-3).	BOLTZ	68
		BOLTZ	69
		BOLTZ	70
NE	= ELECTRON DENSITY (CM-3).	BOLTZ	71
		BOLTZ	72
DNEDT	= INPUT ESTIMATE OF D/DT(NE), THE RATE OF CHANGE OF THE SECONDARY ELECTRON DENSITY (CM-3/SEC). THE BOLTZMANN ANALYSIS DETERMINES D/DT(NE) SELF-CONSISTENTLY IN AN ITERATIVE LOOP, AND EXECUTION MAY BE OPTIMIZED BY A GOOD INITIAL ESTIMATE OF DNEDT. UPON OUTPUT, DNEDT IS THE ACTUAL (SELF-CONSISTENT) VALUE OF D/DT(NE).	BOLTZ	73
		BOLTZ	74
		BOLTZ	75
		BOLTZ	76
		BOLTZ	77
		BOLTZ	78
		BOLTZ	79
SOURCE	= TOTAL RATE OF EXTERNAL SOURCE CREATION OF SECONDARY ELECTRONS AT U = 0 (CM-3/SEC).	BOLTZ	80
		BOLTZ	81
		BOLTZ	82
SBEAM	= TOTAL RATE OF E-BEAM CREATION OF SECONDARY ELECTRONS (CM-3/SEC).	BOLTZ	83
		BOLTZ	84
		BOLTZ	85
		BOLTZ	86
S(I)	= NORMALIZED E-BEAM SOURCE FUNCTION: SBEAM*S(I) = RATE OF GENERATION OF ELECTRONS (/SEC/CM3/EV) IN THE ENERGY RANGE (U,U+DU), WHERE U = X(I). THE TOTAL EXTERNAL POWER DEPOSITION IS: DEPOSIT = SBEAM*INT(DU U S(U)) (EV/SEC/CM3).	BOLTZ	87
		BOLTZ	88
		BOLTZ	89
		BOLTZ	90
		BOLTZ	91
		BOLTZ	92
GAS(K)	= NAME OF THE KTH GAS IN THE MIXTURE.	BOLTZ	93
		BOLTZ	94
FRACT(K)	= RELATIVE FRACTION OF THE KTH GAS IN THE MIXTURE. SUMK(FRACT(K)) NEED NOT EQUAL 1; THE PROGRAM WILL AUTOMATICALLY NORMALIZE THE CONCENTRATIONS, AFTER SORTING THEM INTO DESCENDING NUMERICAL ORDER.	BOLTZ	95
		BOLTZ	96
		BOLTZ	97
		BOLTZ	98
		BOLTZ	99
PROCESS	= ARRAY (DIMENSIONED (4*1)) CONTAINING 4 WORDS PER COLUMN TO PROVIDE A 40-BCD CHARACTER SPECIFICATION FOR EACH OF THE INELASTIC E- COLLISION PROCESSES.	BOLTZ	100
		BOLTZ	101
		BOLTZ	102
		BOLTZ	103
MIX	= NUMBER OF SPECIES IN THE MIXTURE.	BOLTZ	104
		BOLTZ	105
EVCM	= ELECTRIC FIELD (V/CM).	BOLTZ	106
		BOLTZ	107
TMOL	= MOLECULAR TEMPERATURE (DEG K).	BOLTZ	108
		BOLTZ	109
A(I,J)	= SCRATCH ARRAY, DIMENSIONED A(MAX,3) IN THE CALLING PROGRAM, USED FOR STORING CERTAIN COEFFICIENTS THAT DEFINE THE FINITE DIFFERENCE FORM OF THE BOLTZMANN EQUATION (A(I,1), A(I,2), AND A(I,3) ARE USED TO STORE THE THREE MAJOR DIAGONALS DURING THE SOLUTION). LATER, A(I,J) IS USED AS A SCRATCH	BOLTZ	110
		BOLTZ	111
		BOLTZ	112
		BOLTZ	113
		BOLTZ	114
		BOLTZ	115



C		ARRAY FOR CALCULATING NUMEROUS PLASMA PARAMETERS.	BOLTZ	116
C			BOLTZ	117
C	B(I)	= SCRATCH VECTOR, DIMENSIONED B(MAX) IN THE CALLING	BOLTZ	118
C		PROGRAM, USED FOR THE RIGHT HAND SIDE IN THE	BOLTZ	119
C		ITERATIVE SOLUTION OF THE BOLTZMANN EQUATION.	BOLTZ	120
C			BOLTZ	121
C	EPS	= MAXIMUM CHANGE BETWEEN SUCCESSIVE ITERATIONS FOR	BOLTZ	122
C		DEFINITION OF CONVERGENCE FOR THE ELECTRON ENERGY	BOLTZ	123
C		DISTRIBUTION FUNCTION. CONVERGENCE IS DECLARED IF	BOLTZ	124
C			BOLTZ	125
C		$MAX(I)(F(I(N)) - F(I(N-1))) / F(I, N-1) < EPS$	BOLTZ	126
C			BOLTZ	127
C	ITMAX	= MAXIMUM NUMBER OF ITERATIONS ALLOWED TO SOLVE THE	BOLTZ	128
C		BOLTZMANN EQUATION. (IF ITMAX ≤, PROGRAM SETS	BOLTZ	129
C		CONVRG = TRUE, AND IMMEDIATELY TRANSFERS CONTROL	BOLTZ	130
C		TO CALCULATION OF PLASMA PARAMETERS FOR THE INPUT	BOLTZ	131
C		DISTRIBUTION FUNCTION F.	BOLTZ	132
C			BOLTZ	133
C	TMAX	= MAXIMUM CP TIME ALLOWED FOR ATTAINMENT OF CONVER-	BOLTZ	134
C		GENCE OF THE ELECTRON ENERGY DISTRIBUTION FUNCTION	BOLTZ	135
C			BOLTZ	136
C	OUT(I)	= LOGICAL VECTOR OF OUTPUT REQUESTS (I = 1,2,...5):	BOLTZ	137
C			BOLTZ	138
C		1) TABLE OF PLASMA PARAMETERS	BOLTZ	139
C		2) TABLE OF ELECTRON ENERGY DISTRIBUTION	BOLTZ	140
C		3) LOGPLOT OF ENERGY DISTRIBUTION F(U)	BOLTZ	141
C		4) LOGPLOT OF RELATIVE FUNCTION F(U)/FB(U,TE)	BOLTZ	142
C		(FB(U,TE) = BOLTZMANN DISTRIBUTION AT TE.)	BOLTZ	143
C		5) TABLE OF EXCITATION RATES AND ELECTRICAL	BOLTZ	144
C		POWER TRANSFER FOR ALL E(-) PROCESSES.	BOLTZ	145
C			BOLTZ	146
C	KAPTION(4)	= VECTOR CONTAINING A 40 BCD CHARACTER TITLE	BOLTZ	147
C			BOLTZ	148
C		-----	BOLTZ	149
C			BOLTZ	150
C	OUTPUT PARAMETERS --		BOLTZ	151
C			BOLTZ	152
C	VSIG(K,J)	= FORWARD (K = 1) AND REVERSE (K = 2) ELECTRON EX-	BOLTZ	153
C		CITATION RATES (CM3/SEC) FOR THE JTH INELASTIC	BOLTZ	154
C		COLLISION PROCESS.	BOLTZ	155
C			BOLTZ	156
C		VSIG(1,J) = <V(U)Q(U, J)>	BOLTZ	157
C		VSIG(2,J) = <V(U)Q(U,-J)>	BOLTZ	158
C			BOLTZ	159
C		WHERE Q(U,-J) IS THE CROSS SECTION FOR THE REVERSE	BOLTZ	160
C		PROCESS.	BOLTZ	161
C			BOLTZ	162
C	POWER(J)	= NET POWER (ACCOUNTING FOR FORWARD AND REVERSE COL-	BOLTZ	163
C		LISION PROCESSES) PARTITIONED INTO THE JTH INELAS-	BOLTZ	164
C		TIC PROCESS. (WATT/ELECTRON)	BOLTZ	165
C			BOLTZ	166
C	VD	= ELECTRON DRIFT VELOCITY (CM/S).	BOLTZ	167
C			BOLTZ	168
C	MU	= ELECTRON MOBILITY (CM2/V/SEC).	BOLTZ	169
C			BOLTZ	170
C	D	= ELECTRON DIFFUSION COEFFICIENT (CM2/SEC).	BOLTZ	171
C			BOLTZ	172

C	EK	= CHARACTERISTIC ELECTRON ENERGY = D/MU (EV).	BOLTZ	173
C			BOLTZ	174
C	UBAR	= AVERAGE ELECTRON ENERGY (EV).	BOLTZ	175
C			BOLTZ	176
C	TE	= EFFECTIVE ELECTRON TEMPERATURE (DEG K). DEFINED BY	BOLTZ	177
C		(3/2)KTE = UBAR	BOLTZ	178
C			BOLTZ	179
C	DISCH	= E*NE*MU*EVCM**2, TOTAL ELECTRICAL DISCHARGE POWER	BOLTZ	180
C		DENSITY (W/CM3).	BOLTZ	181
C			BOLTZ	182
C	DEPOSIT	= POWER DENSITY FROM EXTERNAL DEPOSITION (INTO ELEC-	BOLTZ	183
C		TRON KINETIC ENERGY) (W/CM3).	BOLTZ	184
C			BOLTZ	185
C	PCOLL	= ELECTRICAL POWER DENSITY INTO INELASTIC COLLISIONS	BOLTZ	186
C		(W/CM3).	BOLTZ	187
C			BOLTZ	188
C	ELASTIC	= ELECTRICAL POWER DENSITY INTO ELASTIC HEATING OF	BOLTZ	189
C		THE MOLECULAR GAS (W/CM3).	BOLTZ	190
C			BOLTZ	191
C	DEDT	= UBAR*ONE/DT = RATE OF CHANGE OF ENERGY DENSITY	BOLTZ	192
C		STORED IN THE ELECTRON GAS (W/CM3).	BOLTZ	193
C			BOLTZ	194
C	NOTE:	IF NE = 0, THE FIVE POWER DENSITIES (DISCH, DE-	BOLTZ	195
C		POSIT, PCOLL, ELASTIC, AND DEDT) ARE EVALUATED FOR	BOLTZ	196
C		UNIT ELECTRON DENSITY (I.E. AS IF NE = 1.0 CM-3),	BOLTZ	197
C		AND THUS, THE EFFECTIVE UNITS ARE W/ELECTRON.	BOLTZ	198
C			BOLTZ	199
C	DNEDT	= D/DT(NE), RATE OF CHANGE OF SECONDARY ELECTRON	BOLTZ	200
C		DENSITY (CM-3/SEC).	BOLTZ	201
C			BOLTZ	202
C	IONIZE	= TOTAL IONIZATION FREQUENCY (SEC-1).	BOLTZ	203
C			BOLTZ	204
C	ATTACH	= TOTAL FREQUENCY FOR ATTACHMENT AND RECOMBINATION.	BOLTZ	205
C			BOLTZ	206
C	DLNEDT	= (1/NE)*D/DT(NE), LOGARITHMIC DERIVATIVE OF NE.	BOLTZ	207
C			BOLTZ	208
C	AMPS	= E*VD = ELECTRICAL CURRENT DENSITY PER UNIT ELEC-	BOLTZ	209
C		TRON DENSITY (A CM).	BOLTZ	210
C			BOLTZ	211
C	F(I)	= NORMALIZED ELECTRON ENERGY DISTRIBUTION FUNCTION	BOLTZ	212
C		(UNITS OF EV**(-3/2)).	BOLTZ	213
C			BOLTZ	214
C	G(I)	= F(I)/F(1)	BOLTZ	215
C			BOLTZ	216
C	B(I)	= F(I)/FBOLTZ(TE,I)	BOLTZ	217
C			BOLTZ	218
C	CONVRGE	= LOGICAL VARIABLE WHICH SPECIFIES SUCCESSFUL CON-	BOLTZ	219
C		VERGENCE OF THE ELECTRON DISTRIBUTION FUNCTION	BOLTZ	220
C		CALCULATIONS.	BOLTZ	221
C			BOLTZ	222
C			BOLTZ	223
C			BOLTZ	224
C			BOLTZ	225
C			BOLTZ	226
C			BOLTZ	227
C			BOLTZ	228
C			BOLTZ	229

DIMENSION XQ(MAX,1), QM(MAX,1), X(1), F(1), G(1), VSIG(2,1),  
 1 U(1), N1(1), N2(1), NEL(1), POWER(1), A(MAX,3), PROCESS(4,1),  
 2 B(1), OUT(1), GAS(1), FRACT(1), NAME(5), FI(5), FORM(15),  
 3 KAPTION(4), NOUT(6), S(1), PCT(6)

REAL NMOL, MU, ME, KT, KB, N1, N2, KTE, NU, NE, LAMBDA, IONIZE,	BOLTZ	230
1 LOG, KVCMTM	BOLTZ	231
LOGICAL CONVRGE, TEST, ERROR, OUT	BOLTZ	232
INTEGER DATE, SKIP, GAS	BOLTZ	233
C	BOLTZ	234
DATA E, PI, KB, ME / 1.602E-12, 3.1415927, 1.38E-16, 9.31E-28 /	BOLTZ	235
DATA NMAX, MASK / 5, 77770000000000000000 /	BOLTZ	236
CALL SECOND(T0)	BOLTZ	237
C	BOLTZ	238
C SORT SPECIES IN DESCENDING ORDER OF CONCENTRATION --	BOLTZ	239
SUM = 0.	BOLTZ	240
DO 37 N = 1,MIX	BOLTZ	241
37 SUM = SUM + FRACT(N)	BOLTZ	242
NAME(1) = NAME(2) = NAME(3) = NAME(4) = NAME(5) = 1H	BOLTZ	243
NGAS = 0	BOLTZ	244
38 FMAX = 0.	BOLTZ	245
DO 2 N = 1,MIX	BOLTZ	246
DO 33 K = 1,NGAS	BOLTZ	247
IF (GAS(N).EQ.NAME(K)) GO TO 2	BOLTZ	248
33 CONTINUE	BOLTZ	249
IF (FRACT(N).LE.FMAX) GO TO 2	BOLTZ	250
FMAX = FRACT(N)	BOLTZ	251
N0 = N	BOLTZ	252
2 CONTINUE	BOLTZ	253
IF (FMAX.LT.1.0E-04*SUM) GO TO 45	BOLTZ	254
NGAS = NGAS+1	BOLTZ	255
NAME(NGAS) = GAS(N0)	BOLTZ	256
FI(NGAS) = FRACT(N0)	BOLTZ	257
IF (NGAS.EQ.NMAX) GO TO 45	BOLTZ	258
GO TO 38	BOLTZ	259
45 PTOT = NMOL*TMOL/0.965E 19	BOLTZ	260
ATM = PTOT/760.	BOLTZ	261
DO 36 N = 1,NGAS	BOLTZ	262
36 FI(N) = FI(N)/SUM	BOLTZ	263
C	BOLTZ	264
FE = NE/NMOL	BOLTZ	265
ESU = 300.*E	BOLTZ	266
ESQ = ESU*ESU	BOLTZ	267
IF (TE.LE.0.) TE = TMOL	BOLTZ	268
KTE = KB*TE	BOLTZ	269
UBAR = 1.5*KTE/E	BOLTZ	270
KT = KB*TMOL/E	BOLTZ	271
CONST = (2.*PI/3.)*NE*ESQ*9.0E 04	BOLTZ	272
DX = X(2)-X(1)	BOLTZ	273
EM = SQRT(2.*E/ME)	BOLTZ	274
M = MESH + 1	BOLTZ	275
MPTS = MESH/2	BOLTZ	276
EMAX = X(M)	BOLTZ	277
C0 = E*1.0E-07	BOLTZ	278
IF (NE.NE.0.) C0 = NE*C0	BOLTZ	279
KAPT = 10HWATT/ELECT	BOLTZ	280
IF (NE.NE.0.) KAPT = 10HWATT/CM3	BOLTZ	281
EN = EVCM/NMOL	BOLTZ	282
ESQ3 = EVCM*EVCM/3.	BOLTZ	283
EP = EVCM/PTOT	BOLTZ	284
KVCMTM = .760*EP	BOLTZ	285
IONIZE = ATTACH = 0.	BOLTZ	286



DLNEDT = 0.	BOLTZ	287
IF (NE,NE.0.) DLNEDT = DNEOT/NE	BOLTZ	288
C = 0.	BOLTZ	289
IF (NE,NE.0.) C = 1./NE	BOLTZ	290
SB = C*SBEAM	BOLTZ	291
S0 = C*SOURCE	BOLTZ	292
SEXT = S0 * SB	BOLTZ	293
C = SB/EM	BOLTZ	294
TEST = SB.NE.0.	BOLTZ	295
CONVRGE = .TRUE.	BOLTZ	296
DO 31 J = 1,NK	BOLTZ	297
31 TEST = TEST.OR.(NEL(J),NE.0)	BOLTZ	298
TEST = TEST.AND.(DLNEDT.EQ.0.)	BOLTZ	299
ELECT = 0.	BOLTZ	300
ITER = 0	BOLTZ	301
TIME = 0.	BOLTZ	302
C INITIAL (COARSE) SOLUTION CONVERGENCE PARAMETER --	BOLTZ	303
EPSILON = 0.01	BOLTZ	304
C	BOLTZ	305
C NORMALIZE THE INPUT DISTRIBUTION FUNCTION --	BOLTZ	306
DO 13 I = 1,M	BOLTZ	307
13 B(I) = SQRT(X(I))*F(I)	BOLTZ	308
CALL SIMPSON (B, MESH/2, DX, FNORM)	BOLTZ	309
DO 14 I = 1,M	BOLTZ	310
A(I,1) = X(I)*S(I)	BOLTZ	311
G(I) = F(I)	BOLTZ	312
14 F(I) = F(I)/FNORM	BOLTZ	313
C	BOLTZ	314
CALL SIMPSON (A, MESH/2, DX, UPLUS)	BOLTZ	315
DEPOSIT = C0*SB*UPLUS	BOLTZ	316
C	BOLTZ	317
C IF ITMAX ≤ 0, THE PROGRAM AUTOMATICALLY SUPPRESSES THE BOLTZMANN	BOLTZ	318
C SOLUTION, AND TRANSFERS DIRECTLY TO CALCULATION OF PLASMA PARA-	BOLTZ	319
C METERS AND EXCITATION RATES BASED UPON THE INPUT CUNCTION F(I).	BOLTZ	320
C	BOLTZ	321
IF (ITMAX.LE.0) GO TO 65	BOLTZ	322
C	BOLTZ	323
C GO TO CALCULATE INITIAL APPROXIMATION TO D/DT(LN NE) --	BOLTZ	324
C IF (TEST) GO TO 65	BOLTZ	325
C	BOLTZ	326
-----	BOLTZ	327
C	BOLTZ	328
C THE BOLTZMANN EQUATION IS REDUCED TO A FINITE DIFFERENCE SYSTEM	BOLTZ	329
C DEFINED OVER AN ENERGY GRID SUBDIVIDED INTO *MESH* INTERVALS. IT	BOLTZ	330
C BECOMES A MATRIX EQUATION AF = B(F). THE MATRIX A, WHICH IS TRI-	BOLTZ	331
C DIAGONAL, INCLUDES ALL OF THE TERMS FROM THE LHS OF THE BOLTZMANN	BOLTZ	332
C EQUATION, AS WELL AS THE DIAGONAL ELEMENTS OF THE RHS INELASTIC	BOLTZ	333
C COLLISION AND ELECTRON-ELECTRON SCATTERING TERMS. AFTER REDUCTION	BOLTZ	334
C OF THE ELECTRON-ELECTRON TERMS TO FINITE DIFFERENCES, ONLY TRI-	BOLTZ	335
C DIAGONAL TERMS RESULT, WITH COEFFICIENTS WHICH ARE EVALUATED USING	BOLTZ	336
C THE PREVIOUS DISTRIBUTION FUNCTION. THE VECTOR B(F) IS A LINEAR	BOLTZ	337
C FUNCTION OF F, COMPOSED OF THE OFF-DIAGONAL ELEMENTS OF THE IN-	BOLTZ	338
C ELASTIC SCATTERING TERM, ALSO EVALUATED USING THE PREVIOUS DISTRI-	BOLTZ	339
C BUTION FUNCTION.	BOLTZ	340
C	BOLTZ	341
C THE BOLTZMANN EQUATION IS WRITTEN	BOLTZ	342
C	BOLTZ	343

C	1/(SQRT(2E/M)*NE)*(SQRT(U)F(U)DNE/DT + DJF(U)/DU + DJEL(U)/DU	BOLTZ	344
C	+ DJEE(U)/DU - SBEAM*S(U)) = ((NU(ION) + SOURCE/NE)*DELTA(U)	BOLTZ	345
C	+ SUMK(U*F(U)*NK*QK(U))),	BOLTZ	346
C		BOLTZ	347
C	WHERE	BOLTZ	348
C		BOLTZ	349
C	J(F) = -(1/3)NE*SQRT(2E/M)*ESQ*U DF/DU /<N QM(U)>	BOLTZ	350
C	J(EL) = - NE*SQRT(2E/M)*USQ<(2ME/M)NQM(U)>*(F + (KT/E)DF/DU)	BOLTZ	351
C	J(EE) = - (2PI/3)*Q**4/E**2*LN(LAMBDA)*NE*NE*SQRT(2E/M)*	BOLTZ	352
C	(PINT(U)*DF/DU + QINT(U)*F(U))	BOLTZ	353
C		BOLTZ	354
C	WHERE	BOLTZ	355
C		BOLTZ	356
C	PINT(U) = 2*INT(DU*U**1.5*F(U)) + 2*U**1.5*INT(DU*F(U))	BOLTZ	357
C	QINT(U) = 3*INT(DU*SQRT(U)*F(U))	BOLTZ	358
C	DEBYE = SQRT(KTE/(4*PI*NE*NE))	BOLTZ	359
C	LAMBDA = DEBYE/RMIN	BOLTZ	360
C	RMIN = ESU**2/(E*UBAR)	BOLTZ	361
C		BOLTZ	362
C	DEFINE QUANTITIES RELATED TO THE DRIVING FIELD AND ELASTIC COLLI-	BOLTZ	363
C	SIONS --	BOLTZ	364
C		BOLTZ	365
C	P(U) = (E**2/3)*U/<N QMOM(U)> + (KT/E)*U**2 <(2ME/M)N QMOM(U)>	BOLTZ	366
C	Q(U) = U**2 <(1ME/M)N QMOM(U)>	BOLTZ	367
C		BOLTZ	368
C	P(I+.5) = (E**2/3)*QM(I,1) + (KT/E)*QM(I,2)	BOLTZ	369
C	Q(I+.5) = QM(I,2)	BOLTZ	370
C		BOLTZ	371
C	DEFINE TERMS RELATED TO ELECTRON-ELECTRON SCATTERING --	BOLTZ	372
C		BOLTZ	373
C	ALPHA(U) = (2PI/3)NE(Q**4/E**2)LOG(LAMBDA)*	BOLTZ	374
C	(PINT(U)/DU + QINT(U)/2)	BOLTZ	375
C	BETA(U) = (2PI/3)NE(Q**4/E**2)LOG(LAMBDA)*	BOLTZ	376
C	(PINT(U)/DU - QINT(U)/2)	BOLTZ	377
C		BOLTZ	378
C	A(I,1) = P(I-.5)/DU - Q(I-.5)/2 + BETA(I-.5)	BOLTZ	379
C	A(I,2) = -(P(I+.5) + P(I-.5))/DU + (Q(I+.5) - Q(I-.5))/2	BOLTZ	380
C	-DU*SUMK(NK U QK(U) + NK*(U + UK)QK(U + UK))	BOLTZ	381
C	- ALPHA(I-.5) - BETA(I+.5) - DU*SQRT(U M/2E)DNE/DT/NE	BOLTZ	382
C	- ALPHA(I-.5) - BETA(I+.5)	BOLTZ	383
C	A(I,3) = P(I+.5)/DU + Q(I+.5)/2 + ALPHA(I+.5)	BOLTZ	384
C		BOLTZ	385
C	B(I) = -DU*SUMK(NK(U*UK)QK(U*UK)F(U*UK) + NK*UQK(U)F(U*UK))	BOLTZ	386
C	- SQRT(M/2E)(DU*S(U)/NE + NU(ION)DELTA(U))	BOLTZ	387
C		BOLTZ	388
C	THE REDUCTION OF THE BOLTZMANN EQUATION TO A TRIDIAGONAL FINITE	BOLTZ	389
C	DIFFERENCE SYSTEM OF EQUATIONS --	BOLTZ	390
C		BOLTZ	391
C	A(I,1)*F(I-1) + A(I,2)*F(I) + A(I,3)*F(I+1) = B(I)	BOLTZ	392
C		BOLTZ	393
C	I = 2,3,...,M. THE SET IS COMPLETED BY THE CONDITION THAT F(1)	BOLTZ	394
C	CAN BE APPROXIMATED AS (E.G.) F(1) = F(2).	BOLTZ	395
C		BOLTZ	396
C	IN GENERAL, DIRECT INVERSION OF THE LINEAR SYSTEM IS NOT PRACTICAL	BOLTZ	397
C	EITHER FROM THE STANDPOINT OF CORE STORAGE OR MATRIX INVERSION.	BOLTZ	398
C	INSTEAD, ADVANTAGE IS TAKEN OF THE FACT THAT A IS SPARSE (I.E. ALL	BOLTZ	399
C	ELEMENTS EXCEPT FOR THREE DIAGONALS ARE ZERO), AND SOLUTION IS	BOLTZ	400

C	ATTEMPTED BY AN ITERATIVE APPROACH: $A(F(N)) * F(N+1) = B(F(N))$ .	BOLTZ	401
C		BOLTZ	402
C	-----	BOLTZ	403
C		BOLTZ	404
	40 TRANS = DLNEDT/EM	BOLTZ	405
C	THE FIRST EQUATION IS $F(1) - F(2) = 0$ --	BOLTZ	406
	A(1,1) = 0.	BOLTZ	407
	A(1,2) = 1.0	BOLTZ	408
	A(1,3) = - 1.0	BOLTZ	409
	Q1 = QM(1,2)/2.	BOLTZ	410
	P1 = (KT*QM(1,2) + ESQ3*QM(1,1))/DX	BOLTZ	411
	DO 1 I = 2,M	BOLTZ	412
	Q2 = QM(1,2)/2.	BOLTZ	413
	P2 = (KT*QM(1,2) + ESQ3*QM(1,1))/DX	BOLTZ	414
	DIAG = 0.	BOLTZ	415
	DO 4 J = 1,NK	BOLTZ	416
	DIAG = DIAG + N1(J)*XQ(I,J)	BOLTZ	417
	JU = UJ = U(J)/DX	BOLTZ	418
	IJ = I+JU	BOLTZ	419
	IF (IJ.GT.M) GO TO 4	BOLTZ	420
	IF (IJ.LT.1) GO TO 4	BOLTZ	421
	DIAG = DIAG + N2(J)*XQ(IJ,J)	BOLTZ	422
	4 CONTINUE	BOLTZ	423
	ROOT = SQRT(X(I))	BOLTZ	424
	A(I,1) = P1 - Q1	BOLTZ	425
	A(I,2) = - P2 - P1 + Q2 - Q1 - (DIAG + ROOT*TRANS)*DX	BOLTZ	426
	A(I,3) = P2 + Q2	BOLTZ	427
	P1 = P2	BOLTZ	428
	1 Q1 = Q2	BOLTZ	429
C		BOLTZ	430
	IF (NE.EQ.0.) GO TO 10	BOLTZ	431
C	INTEGRAL OF F(U) (FOR E-E SCATTERING) --	BOLTZ	432
	CALL SIMPSON (F, MESH/2, 1., FINT)	BOLTZ	433
C		BOLTZ	434
C	-----	BOLTZ	435
C		BOLTZ	436
C	CALCULATE THE RHS VECTOR B, WHICH CONTAINS ALL OF THE OFF-DIAGONAL	BOLTZ	437
C	PARTS OF THE INELASTIC COLLISION TERMS, DEFINED IN TERMS OF THE	BOLTZ	438
C	PRESENT DISTRIBUTION VECTOR F(N):	BOLTZ	439
C		BOLTZ	440
	10 IF (ITER.EQ.ITMAX) GO TO 98	BOLTZ	441
	CALL SECOND (TIME)	BOLTZ	442
	TIME = TIME-T0	BOLTZ	443
	IF (TIME.GT.TMAX) GO TO 98	BOLTZ	444
	ITER = ITER+1	BOLTZ	445
	B(1) = 0.	BOLTZ	446
	DO 5 I = 2,M	BOLTZ	447
	OFF = 0.	BOLTZ	448
	DO 8 J = 1,NK	BOLTZ	449
	JU = U(J)/DX	BOLTZ	450
	IF (NEL(J).LT.0) GO TO 8	BOLTZ	451
C		BOLTZ	452
C	COLLISIONS (OF THE FIRST KIND) IN WHICH ELECTRONS ARE	BOLTZ	453
C	SCATTERED INTO ENERGY X(I), LOSING ENERGY U(J) --	BOLTZ	454
	IJ = I+JU	BOLTZ	455
	IF (IJ.GT.M) GO TO 6	BOLTZ	456
	IF (IJ.LT.1) GO TO 6	BOLTZ	457



	OFF = OFF + XQ(IJ,J)*F(IJ)*N1(J)	BOLTZ	458
C		BOLTZ	459
	6 IF (NEL(J).GT.0) GO TO 8	BOLTZ	460
C		BOLTZ	461
C	COLLISIONS (OF THE SECOND KIND) IN WHICH ELECTRONS ARE	BOLTZ	462
C	SCATTERED INTO ENERGY X(I), GAINING ENERGY U(J) --	BOLTZ	463
	IJ = I-JU	BOLTZ	464
	IF (IJ.LT.1) GO TO 8	BOLTZ	465
	IF (IJ.GT.M) GO TO 8	BOLTZ	466
	OFF = OFF + XQ(I,J)*F(IJ)*N2(J)	BOLTZ	467
C		BOLTZ	468
	R CONTINUE	BOLTZ	469
	5 B(1) = - (C*S(1) + OFF)*DX	BOLTZ	470
C		BOLTZ	471
C	PUT THE DELTA-FUNCTION SINGULARITY ARISING FROM SECONDARY IONIZA-	BOLTZ	472
C	TION AND EXTERNAL CREATION OF ELECTRONS AT ZERO ENERGY AT U = DX	BOLTZ	473
C	(I.E., AT I = 2) --	BOLTZ	474
	B(2) = B(2) - (IONIZE + S0)/EM	BOLTZ	475
C		BOLTZ	476
C	-----	BOLTZ	477
C		BOLTZ	478
C	SOLVE THE LINEAR EQUATION: A(F(N))*F(N+1) = B(F(N)) --	BOLTZ	479
C		BOLTZ	480
C	THIS LOOP FOR SOLVING A*F(N+1) = B WILL ULTIMATELY PLACE THE SOLU-	BOLTZ	481
C	TION F(N+1) INTO THE VECTOR G, AND DOES NOT DESTROY THE MATRIX A	BOLTZ	482
C	CONSTRUCTED ABOVE. THE SOLUTION WORKS DOWNWARD (ANNIHILATING THE	BOLTZ	483
C	LOWER DIAGONAL) BY MEANS OF ELEMENTARY ROW OPERATIONS, FOLLOWED	BOLTZ	484
C	WITH BACK SUBSTITUTION UPWARDS.	BOLTZ	485
C		BOLTZ	486
C	IN THE DOWNWARD ANNIHILATION OF THE LOWER DIAGONAL, THE VECTOR	BOLTZ	487
C	G(I) IS TEMPORARILY USED TO STORE THE COEFFICIENTS OF F(I+1) --	BOLTZ	488
C		BOLTZ	489
	G(1) = A(1,3)/A(1,2)	BOLTZ	490
	B(1) = B(1)/A(1,2)	BOLTZ	491
	F1 = -F(1)/2.	BOLTZ	492
	F2 = F3 = 0.	BOLTZ	493
	ELECT = PNEW = 0.	BOLTZ	494
	A2 = B2 = 0.	BOLTZ	495
	XHALF = DX/2.	BOLTZ	496
	IF (NE.EQ.0.) GO TO 35	BOLTZ	497
C	DEBYE LENGTH FOR PLASMA --	BOLTZ	498
	DEBYE = SQRT(KTE/(4.*PI*NE*ESQ))	BOLTZ	499
C	CLASSICAL DISTANCE OF CLOSEST APPROACH --	BOLTZ	500
	RMIN = 300.*ESU/UBAR	BOLTZ	501
	LAMBDA = DEBYE/RMIN	BOLTZ	502
	LOG = ALOG(LAMBDA)	BOLTZ	503
35	DO 26 I = 1,M	BOLTZ	504
	A1 = A2	BOLTZ	505
	B1 = B2	BOLTZ	506
	IF (NE.EQ.0.) GO TO 20	BOLTZ	507
	XROOT = SQRT(X(I))	BOLTZ	508
	X32 = X(I)*XROOT	BOLTZ	509
	F1 = F1 + F(I)	BOLTZ	510
	F2 = F2 + XROOT*F(I)	BOLTZ	511
	F3 = F3 + X32*F(I)	BOLTZ	512
	X32 = XHALF**1.5	BOLTZ	513
	XHALF = XHALF + DX	BOLTZ	514

C	PU = PINT(U)/DU, AND QU = QINT(U)/2.	BOLTZ	515
	PU = F3 + X32*(FINT-F1)	BOLTZ	516
	PU = PU + PU	BOLTZ	517
	QU = 1.5*F2*DX	BOLTZ	518
	A2 = CONST*(PU + QU)*LOG	BOLTZ	519
	B2 = CONST*(PU - QU)*LOG	BOLTZ	520
	DF = FAVG2 = 0.	BOLTZ	521
	IF (I.LT.M) DF = F(I+1) - F(I)	BOLTZ	522
	IF (I.LT.M) FAVG2 = F(I+1) + F(I)	BOLTZ	523
	POLD = PNEW	BOLTZ	524
	PNEW = (PU*DF + QU*FAVG2)*LOG	BOLTZ	525
	ELECT = ELECT + 0.5*(POLD + PNEW)	BOLTZ	526
20	IF (I.EQ.1) GO TO 26	BOLTZ	527
	B(I) = B(I) - (A(I,1) + B1)*B(I-1)	BOLTZ	528
	G(I) = A(I,3) + A2	BOLTZ	529
	Q = (A(I,2) - A1 - B2) - (A(I,1) + B1)*G(I-1)	BOLTZ	530
	B(I) = B(I)/Q	BOLTZ	531
	G(I) = G(I)/Q	BOLTZ	532
26	CONTINUE	BOLTZ	533
C		BOLTZ	534
C	UPWARD BACK SUBSTITUTION, WITH NEW SOLUTION INTO G --	BOLTZ	535
C		BOLTZ	536
	G(M) = B(M)	BOLTZ	537
	DO 3 J = 1,MESH	BOLTZ	538
	I = M-J	BOLTZ	539
	3 G(I) = B(I) - G(I)*G(I+1)	BOLTZ	540
C		BOLTZ	541
C	-----	BOLTZ	542
C		BOLTZ	543
C	RENORMALIZE THE NEW DISTRIBUTION FUNCTION, AND TEST FOR	BOLTZ	544
C	CONVERGENCE --	BOLTZ	545
C		BOLTZ	546
	DO 25 I = 1,M	BOLTZ	547
25	B(I) = SQRT(X(I))*G(I)	BOLTZ	548
C		BOLTZ	549
	CALL SIMPSON (B, MESH/2, DX, FNORM)	BOLTZ	550
	CALL SIMPSON (G, MESH/2, 1.0, GINT)	BOLTZ	551
	FINT = GINT/FNORM	BOLTZ	552
C		BOLTZ	553
	DELTA = 0.	BOLTZ	554
	ERROR = .FALSE.	BOLTZ	555
	CONVRGE = .TRUE.	BOLTZ	556
	DO 11 I = 1,M	BOLTZ	557
	FOLD = F(I)	BOLTZ	558
	F(I) = G(I)/FNORM	BOLTZ	559
	FNEW = F(I)	BOLTZ	560
	B(I) = B(I)*X(I)	BOLTZ	561
	F1 = ABS(FOLD)	BOLTZ	562
	F2 = ABS(FNEW)	BOLTZ	563
	FMAX = F1	BOLTZ	564
	IF (F2.GT.F1) FMAX = F2	BOLTZ	565
	ERROR = ERROR.OR.(FNEW.LT.0.)	BOLTZ	566
	ETA = 0.	BOLTZ	567
	IF ((FMAX.AND.MASK).NE.0) ETA = ABS(FNEW-FOLD)/FMAX	BOLTZ	568
	IF (ETA.GT.DELTA) DELTA = ETA	BOLTZ	569
	TEST = ETA.LE.EPSILON	BOLTZ	570
11	CONVRGE = CONVRGE.AND.TEST	BOLTZ	571

C	CUNVRGE = CONVRGE.AND.(.NOT.ERROR)	BOLTZ	572
C	CALL SIMPSON (B, MESH/2, DX, UBAR)	BOLTZ	573
C	UBAR = UBAR/FNORM	BOLTZ	574
C	KTE = (2./3.)*E*UBAR	BOLTZ	575
C	TE = KTE/KB	BOLTZ	576
C	IF (.NOT.CONVRGE) GO TO 10	BOLTZ	577
C	-----	BOLTZ	578
C		BOLTZ	579
C		BOLTZ	580
C	CALCULATE THE FORWARD AND REVERSE COLLISION RATES VSIG(1,J) AND	BOLTZ	581
C	VSIG(2,J) (CM3/SEC) AND NET POWER BALANCE POWER(J) (WATT/ELECTRON)	BOLTZ	582
C	FOR EACH OF THE INELASTIC PROCESSES J --	BOLTZ	583
C		BOLTZ	584
C	65 DLNEDT = DLNEDT	BOLTZ	585
C	PCOLL = 0.	BOLTZ	586
C	IONIZE AND ATTACH ARE THE IONIZATION AND ATTACHMENT FREQUENCIES--	BOLTZ	587
C	IONIZE = ATTACH = 0.	BOLTZ	588
C	DO 16 J = 1,NK	BOLTZ	589
C	FORWARD = REVERSE = 0.	BOLTZ	590
C	JU = UJ = U(J)/DX	BOLTZ	591
C	DO 15 I = 1,M	BOLTZ	592
C	A(I,1) = F(I)*XQ(I,J)	BOLTZ	593
C	A(I,2) = 0.	BOLTZ	594
C	IJ = I+JU	BOLTZ	595
C	IF (IJ,LT.1) GO TO 15	BOLTZ	596
C	IF (IJ,GT.M) GO TO 15	BOLTZ	597
C	A(I,2) = F(I)*XQ(IJ,J)	BOLTZ	598
C	15 A(I,3) = X(I)*A(I,1)	BOLTZ	599
C		BOLTZ	600
C	CALL SIMPSON (A(1,1), MPTS, DX, VSIG(1,J))	BOLTZ	601
C	CALL SIMPSON (A(1,2), MPTS, DX, VSIG(2,J))	BOLTZ	602
C		BOLTZ	603
C	NOTE: EM = SQRT(2*(1.602E-12)/9.31E-28) = 5.866E 07 EV**(-1/2)CM/S.	BOLTZ	604
C	UNITS OF F(U)UQ(U)DU ARE EV**(1/2)CM2, SO RESULT FOR <VSIG> WILL	BOLTZ	605
C	BE IN UNITS OF CM3/SEC.	BOLTZ	606
C		BOLTZ	607
C	VSIG(1,J) = EM*VSIG(1,J)	BOLTZ	608
C	VSIG(2,J) = EM*VSIG(2,J)	BOLTZ	609
C		BOLTZ	610
C	IF (NEL(J).LT.0) GO TO 24	BOLTZ	611
C	FORWARD = U(J)*VSIG(1,J)	BOLTZ	612
C	REVERSE = U(J)*VSIG(2,J)	BOLTZ	613
C	GO TO 27	BOLTZ	614
C		BOLTZ	615
C	24 CALL SIMPSON (A(1,3), MPTS, DX, FORWARD)	BOLTZ	616
C	FORWARD = EM*FORWARD	BOLTZ	617
C		BOLTZ	618
C	27 POWER(J) = C0*(N1(J)*FORWARD - N2(J)*REVERSE)	BOLTZ	619
C		BOLTZ	620
C	FREQ = N1(J)*VSIG(1,J)	BOLTZ	621
C	IF (NEL(J).EQ.+1) IONIZE = IONIZE + FREQ	BOLTZ	622
C	IF (NEL(J).EQ.-1) ATTACH = ATTACH + FREQ	BOLTZ	623
C		BOLTZ	624
C	16 PCOLL = PCOLL + POWER(J)	BOLTZ	625
C	DLNEDT = SEXY + IONIZE - ATTACH	BOLTZ	626
C	IF (ITMAX.LE.0) GO TO 60	BOLTZ	627
C		BOLTZ	628



C	D/DT(NE) IS DETERMINED SELF-CONSISTENTLY BY AN ITERATIVE LOOP. IF	BOLTZ	629
C	THE CHANGE IN THE (LOGARITHMIC) DERIVATIVE OF NE BECOMES $\leq 2\%$ , NO	BOLTZ	630
C	FURTHER ITERATION IS REQUIRED --	BOLTZ	631
C	IF (ABS(DLNEDT-DLNEDT0).GT.ABS(DLNEDT)/50.) GO TO 40	BOLTZ	632
C	IF (EPS.GE.EPSILON) GO TO 60	BOLTZ	633
C	FINE TUNING OF CALCULATION --	BOLTZ	634
C	EPSILON = EPS	BOLTZ	635
C	GO TO 40	BOLTZ	636
C	DETERMINE CP TIME REQUIRED FOR OBTAINING F(U) --	BOLTZ	637
C	60 CALL SECOND (TIME)	BOLTZ	638
C	TIME = TIME-T0	BOLTZ	639
C	-----	BOLTZ	640
C	XBAR = -DX/2.	BOLTZ	641
C	DO 9 I = 1,M	BOLTZ	642
C	G(I) = F(I)/F(1)	BOLTZ	643
C	B(I) = B(I)*X(I)	BOLTZ	644
C	CALL INTERP (1, XBAR, QMAVG, X, QM(1,1), 1, M)	BOLTZ	645
C	A(I,1) = F(I)*X(I)*X(I)/QMAVG	BOLTZ	646
C	IF (F(I).GT.1.E-20*F(1)) MP = I	BOLTZ	647
C	9 XBAR = XBAR + DX	BOLTZ	648
C	F HAS BEEN NORMALIZED TO $\text{INT}(DU*\text{SQRT}(U)*F(U)) = 1$ , AND $G(I) =$	BOLTZ	649
C	$F(I)/F(1)$ .	BOLTZ	650
C	ELASTIC COLLISION FREQUENCY NU:	BOLTZ	651
C	CALL SIMPSON (A, MPTS, DX, NU)	BOLTZ	652
C	NU = EM*NU	BOLTZ	653
C	CALCULATE AVERAGE ENERGY AND EFFECTIVE TEMPERATURE --	BOLTZ	654
C	IF (ITER.GT.0) GO TO 75	BOLTZ	655
C	CALL SIMPSON (B, MESH/2, DX, UBAR)	BOLTZ	656
C	UBAR = UBAR/FNORM	BOLTZ	657
C	KTE = (2./3.)*E*UBAR	BOLTZ	658
C	TE = KTE/KB	BOLTZ	659
C	-----	BOLTZ	660
C	CALCULATE DIFFUSION COEFFICIENT D, MOBILITY MU, CHARACTERISTIC	BOLTZ	661
C	ENERGY EK = D/MU, DRIFT VELOCITY VD = MU*EVCM, AND THE FORWARD AND	BOLTZ	662
C	REVERSE ELASTIC POWER TRANSFER --	BOLTZ	663
C	75 F1 = F(1)	BOLTZ	664
C	DO 12 I = 1,MESH	BOLTZ	665
C	F0 = F1	BOLTZ	666
C	F1 = F(I+1)	BOLTZ	667
C	FAVG = (F0 + F1)/2.	BOLTZ	668
C	DF = F1-F0	BOLTZ	669
C	A(I,1) = QM(I,1)*FAVG	BOLTZ	670
C	A(I,2) = QM(I,1)*DF	BOLTZ	671
C	A(I,3) = QM(I,2)*FAVG	BOLTZ	672
C		BOLTZ	673
C		BOLTZ	674
C		BOLTZ	675
C		BOLTZ	676
C		BOLTZ	677
C		BOLTZ	678
C		BOLTZ	679
C		BOLTZ	680
C		BOLTZ	681
C		BOLTZ	682
C		BOLTZ	683
C		BOLTZ	684
C		BOLTZ	685

12	B(1) = QM(I,2)*DF	BOLTZ	686
	A(M,1) = QM(M,1)*F1	BOLTZ	687
	A(M,2) = QM(M,1)*DF	BOLTZ	688
	A(M,3) = QM(M,2)*F1	BOLTZ	689
	B(M) = QM(M,2)*DF	BOLTZ	690
C		BOLTZ	691
C	-----	BOLTZ	692
C		BOLTZ	693
C	DIFFUSION COEFFICIENT D (CM2/SEC) --	BOLTZ	694
C	CALL SIMPSON (A(1,1), MPTS, DX, D)	BOLTZ	695
	D = EM*D/3.	BOLTZ	696
C		BOLTZ	697
C	MOBILITY MU (CM2/VOLT/SEC) --	BOLTZ	698
C		BOLTZ	699
C	CALL SIMPSON (A(1,2), MPTS, 1.0, MU)	BOLTZ	700
C	INTEGRATION-BY-PARTS CORRECTION TERM --	BOLTZ	701
	MU = MU - EMAX*A(M,2)/DX	BOLTZ	702
	MU = - EM*MU/3.	BOLTZ	703
C		BOLTZ	704
	EK = D/MU	BOLTZ	705
	VD = MU*EVCM	BOLTZ	706
	AMPS = E*1.0E-07*VD	BOLTZ	707
	COND = E*1.0E-07*MU	BOLTZ	708
	RMH = 1./COND	BOLTZ	709
	CONDUCT = NE*COND	BOLTZ	710
	DNEDT = NE*DLNEDT	BOLTZ	711
C		BOLTZ	712
C	-----	BOLTZ	713
C		BOLTZ	714
C	FORWARD (PF) AND REVERSE (PR) ELASTIC POWER --	BOLTZ	715
C		BOLTZ	716
C	CALL SIMPSON (A(1,3), MPTS, DX, PF)	BOLTZ	717
	PF = C0*EM*PF	BOLTZ	718
C		BOLTZ	719
C	CALL SIMPSON (B, MPTS, 1.0, PR)	BOLTZ	720
	PR = C0*EM*KT*PR	BOLTZ	721
C		BOLTZ	722
C	ELASTIC = PF + PR	BOLTZ	723
C		BOLTZ	724
C	ELECTRIC DISCHARGE POWER DENSITY --	BOLTZ	725
C	DISCH = C0*MU*EVCM*EVCM	BOLTZ	726
C		BOLTZ	727
C	RATE OF CHANGE OF STORED ELECTRON KINETIC ENERGY --	BOLTZ	728
C	DEDT = C0*UBAR*DLNEDT	BOLTZ	729
C		BOLTZ	730
C	PTOTAL = DISCH + DEPOSIT	BOLTZ	731
	PWR = PCOLL + ELASTIC + DEDT	BOLTZ	732
C	ELECTRON-ELECTRON SCATTERING POWER DISCREPANCY --	BOLTZ	733
	ELECT = C0*EM*CONST*DX*ELECT	BOLTZ	734
C		BOLTZ	735
	IF (PTOTAL.EQ.0.) GO TO S3	BOLTZ	736
	P = PTOTAL/100.	BOLTZ	737
	PCT(1) = ELASTIC/P	BOLTZ	738
	PCT(2) = PCOLL/P	BOLTZ	739
	PCT(3) = DEDT/P	BOLTZ	740
	PCT(4) = ELECT/P	BOLTZ	741
		BOLTZ	742

	PCT(5) = DISCH/P	BOLTZ	743
	PCT(6) = DEPOSIT/P	BOLTZ	744
	PERCENT = PCT(1) + PCT(2) + PCT(3) + PCT(4)	BOLTZ	745
	PBAL = PERCENT - 100.	BOLTZ	746
	GO TO 54	BOLTZ	747
C		BOLTZ	748
53	PMAX = AMAX1(ABS(ELASTIC), ABS(PCOLL), ABS(DEDT))	BOLTZ	749
	P = ELASTIC + ELECT + PCOLL + DEDT	BOLTZ	750
	PBAL = 100.*P/PMAX	BOLTZ	751
C		BOLTZ	752
54	FB = 2./(KTE/E)**1.5/SQRT(PI)	BOLTZ	753
	EX = EXP(-E*DX/KTE)	BOLTZ	754
	DO 19 I = 1,M	BOLTZ	755
	B(I) = F(I)/FB	BOLTZ	756
	19 FB = FR*EX	BOLTZ	757
C		BOLTZ	758
C	-----	BOLTZ	759
C		BOLTZ	760
C	TABLE OF PLASMA PARAMETERS (UBAR, TE, VD, D, ... ETC.) --	BOLTZ	761
C		BOLTZ	762
	NG1 = NGAS-1	BOLTZ	763
	IF (.NOT.OUT(1)) GO TO 21	BOLTZ	764
	NX = 68-6*NGAS	BOLTZ	765
	ENCODE (150,250,FORM) NX, NG1, NG1	BOLTZ	766
	WRITE (6,350) KAPTION	BOLTZ	767
	IF (NGAS.EQ.1) WRITE (6,210) NAME(1)	BOLTZ	768
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I=1,NGAS), (F(I), I=1,NGAS)	BOLTZ	769
C		BOLTZ	770
	WRITE (6,100) EVCM, TMOL, NMOL, PTOT, ATM, EN, EP, KVCMATM, UBAR,	BOLTZ	771
1	TE, D, MU, EK, VD, COND, CONDUCT, RHO, AMPS, DISCH, KAPT,	BOLTZ	772
2	DEPOSIT, KAPT, PTOTAL, KAPT, PCOLL, KAPT, ELASTIC, KAPT, DEDT,	BOLTZ	773
3	KAPT, PWR, KAPT, NU, NE, FE, S0, SB, IONIZE, ATTACH, DLNEDT,	BOLTZ	774
4	DNEDT	BOLTZ	775
C		BOLTZ	776
	WRITE (6,380) MESH, EMAX, DX	BOLTZ	777
C		BOLTZ	778
C	-----	BOLTZ	779
C		BOLTZ	780
C	TABULAR OUTPUT OF ELECTRON ENERGY DISTRIBUTION FUNCTION --	BOLTZ	781
C		BOLTZ	782
21	NX = 58 - 6*NGAS	BOLTZ	783
	ENCODE (150,220,FORM) NX, NG1, NG1	BOLTZ	784
	IF (.NOT.OUT(2)) GO TO 22	BOLTZ	785
	MULT = (MESH-1)/100 + 1	BOLTZ	786
	DU = MULT*DX	BOLTZ	787
	WRITE (6,350) KAPTION	BOLTZ	788
	WRITE (6,200) DU, X(M), MESH, DX	BOLTZ	789
	WRITE (6,330) EN, EP, KVCMATM	BOLTZ	790
	IF (FE.GT.0.) WRITE (6,320) FE, NE	BOLTZ	791
	IF (NGAS.EQ.1) GO TO 17	BOLTZ	792
	WRITE (6,FORM) (NAME(I), I = 1,NGAS), (F(I), I=1,NGAS), TMOL	BOLTZ	793
	GO TO 7	BOLTZ	794
17	WRITE (6,260) NAME(1), TMOL	BOLTZ	795
7	WRITE (6,290)	BOLTZ	796
	WRITE (6,240) (X(I), F(I), I = 1,M,MULT)	BOLTZ	797
	WRITE (6,160)	BOLTZ	798
	IF (ITER.GT.0) WRITE (6,180) TIME, ITER, DELTA, PBAL	BOLTZ	799



	WRITE (6,280) DATE	BOLTZ	800
C	22 IF (.NOT.OUT(3)) GO TO 23	BOLTZ	801
C	-----	BOLTZ	802
C		BOLTZ	803
C	PLOT THE DISTRIBUTION FUNCTION F(U)/F(0) --	BOLTZ	804
C		BOLTZ	805
	WRITE (6,110) F(1)	BOLTZ	806
	WRITE (6,330) EN, EP, KVCMATM	BOLTZ	807
	IF (FE.GT.0.) WRITE (6,320) FE, NE	BOLTZ	808
	IF (NGAS.EQ.1) WRITE (6,260) NAME(1), TMOL	BOLTZ	809
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS),	BOLTZ	810
	1 (F1(I), I = 1,NGAS), TMOL	BOLTZ	811
C		BOLTZ	812
	DF = 2.	BOLTZ	813
	IF (G(MP).GT.1.E-10) DF = 1.	BOLTZ	814
	F0 = -10.*DF	BOLTZ	815
	CALL PXLOGY (MAX, MP, 1, G, F0, DF, X, 0., 0., .TRUE., .FALSE.,	BOLTZ	816
	1, .TRUE., .TRUE., .TRUE., TITLE, 1, 0)	BOLTZ	817
	WRITE (6,150) DATE	BOLTZ	818
C	-----	BOLTZ	819
C		BOLTZ	820
C	BOLTZMANN DISTRIBUTION FUNCTION WITH EFFECTIVE TEMPERATURE TE IS	BOLTZ	821
C	GIVEN BY FB(U,TE) = 2(KTE/E)**(-3/2)/SQRT(PI)*EXP(-E*U/KTE). PLOT	BOLTZ	822
C	F(U)/FB(U,TE) --	BOLTZ	823
C		BOLTZ	824
C	23 IF (.NOT.OUT(4)) GO TO 30	BOLTZ	825
C		BOLTZ	826
	WRITE (6,120) TE	BOLTZ	827
	WRITE (6,330) EN, EP, KVCMATM	BOLTZ	828
	IF (FE.GT.0.) WRITE (6,320) FE, NE	BOLTZ	829
	IF (NGAS.EQ.1) WRITE (6,260) NAME(1), TMOL	BOLTZ	830
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS),	BOLTZ	831
	1 (F1(I), I = 1,NGAS), TMOL	BOLTZ	832
C		BOLTZ	833
	CALL PXLOGY (MAX, MP, 1, B, -4.0, .5, X, 0., 0., .TRUE., .FALSE.,	BOLTZ	834
	1, .TRUE., .TRUE., .TRUE., TITLE, 1, 0)	BOLTZ	835
C		BOLTZ	836
	WRITE (6,150) DATE	BOLTZ	837
C	-----	BOLTZ	838
C		BOLTZ	839
C	TABULAR PLOT OF COLLISION RATES, POWER TRANSFER --	BOLTZ	840
C		BOLTZ	841
C	30 IF (.NOT.OUT(5)) GO TO 99	BOLTZ	842
	SKIP = 5H(1H1)	BOLTZ	843
	LSKIP = 1H	BOLTZ	844
	IF (SB.EQ.0.) LSKIP = 1H.	BOLTZ	845
	L = 15	BOLTZ	846
	LINE = LL = 0	BOLTZ	847
	DO 18 J = 1,NK	BOLTZ	848
	IF (LL.NE.LINE) GO TO 28	BOLTZ	849
	J1 = L - (NK-J+1)	BOLTZ	850
	IF (J1.GT.7) J1 = 7	BOLTZ	851
	IF (J1.GE.1) ENCODE (10,300,SKIP) J1	BOLTZ	852
		BOLTZ	853
		BOLTZ	854
		BOLTZ	855
		BOLTZ	856

WRITE (6,SKIP)	BOLTZ	857
WRITE (6,270) KAPTION	BOLTZ	858
WRITE (6,330) EN, EP, KVCMTM	BOLTZ	859
IF (NE.GT.0.) WRITE (6,320) FE, NE	BOLTZ	860
IF (NGAS.EQ.1) WRITE (6,260) NAME(1), TMOL	BOLTZ	861
IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS),	BOLTZ	862
1 (FI(I), I=1,NGAS), TMOL	BOLTZ	863
WRITE (6,140) UBAR, TE, VD, MU	BOLTZ	864
WRITE (6,230) KAPT	BOLTZ	865
IF (LINE.GT.0) GO TO 28	BOLTZ	866
WRITE (6,170) ELASTIC	BOLTZ	867
IF (PTOTAL.NE.0.) WRITE (6,420) PCT(1)	BOLTZ	868
IF (SB.NE.0.) WRITE (6,430) SB	BOLTZ	869
IF (SO.NE.0.) WRITE (6,390) LSKIP, SO	BOLTZ	870
28 IF (N1(J)*N2(J).EQ.0.) GO TO 18	BOLTZ	871
LINE = LINE+1	BOLTZ	872
FREQ = NEL(J)*N1(J)*VSIG(1,J)	BOLTZ	873
IF (PTOTAL.NE.0.) PCTJ = POWER(J)/P	BOLTZ	874
DO 29 I = 1,6	BOLTZ	875
29 NOUT(I) = 1H	BOLTZ	876
IF (N1(J).NE.0.) ENCODE (10,190,NOUT(1)) N1(J)	BOLTZ	877
IF (N2(J).NE.0.) ENCODE (10,190,NOUT(2)) N2(J)	BOLTZ	878
IF (NEL(J).GE.0) ENCODE (10,360,NOUT(3)) U(J)	BOLTZ	879
IF (NEL(J).NE.0) ENCODE (10,190,NOUT(4)) FREQ	BOLTZ	880
ENCODE (10,190,NOUT(5)) VSIG(1,J)	BOLTZ	881
IF (NEL(J).EQ.0) ENCODE (10,190,NOUT(6)) VSIG(2,J)	BOLTZ	882
C WRITE (6,130) LINE, (PROCESS(I,J), I = 1,4), NOUT, POWER(J)	BOLTZ	883
IF (PTOTAL.NE.0.) WRITE (6,420) PCTJ	BOLTZ	884
C	BOLTZ	885
LL = L*(LINE/L)	BOLTZ	886
IF (LL.EQ.LINE) WRITE (6,160)	BOLTZ	887
18 CONTINUE	BOLTZ	888
IF (LL.NE.LINE) WRITE (6,160)	BOLTZ	889
PWR = PWR + ELECT	BOLTZ	890
IF (PTOTAL.EQ.0.) GO TO 55	BOLTZ	891
WRITE (6,370) DISCH, KAPT, PCT(5), PCOLL, PCT(2), DEPOSIT, KAPT,	BOLTZ	892
1 PCT(6), ELASTIC, PCT(1), DEDT, PCT(3)	BOLTZ	893
IF (NE.GT.0.) WRITE (6,310) ELECT, PCT(4)	BOLTZ	894
WRITE (6,340) PTOTAL, KAPT, PWR, PERCENT	BOLTZ	895
GO TO 56	BOLTZ	896
55 WRITE (6,400) PCOLL, KAPT, ELASTIC, KAPT, DEDT, KAPT	BOLTZ	897
IF (NE.GT.0.) WRITE (6,310) ELECT	BOLTZ	898
WRITE (6,410) PWR, KAPT	BOLTZ	899
56 IF (ITER.GT.0) WRITE (6,180) TIME, ITER, DELTA, PBAL	BOLTZ	900
WRITE (6,280) DATE	BOLTZ	901
C	BOLTZ	902
GO TO 99	BOLTZ	903
C	BOLTZ	904
C	BOLTZ	905
C ----- FORMAT STATEMENTS -----	BOLTZ	906
C	BOLTZ	907
100 FORMAT (/60X*PLASMA PARAMETERS*//48X,42(1H-)//50X*E (FIELD)*4X* = *	BOLTZ	908
11PE11.4,3X*VOLT/CM*//50X*TMOL*8X* = *0PF8.0,6X*DEG K*/50X*NMOL*8X	BOLTZ	909
2* = *1PE11.4,3X*CM-3*/50X*PTOT*8X* = *0PF9.1,5X*TORR*/62X* = *F8.2	BOLTZ	910
3,6X*ATM*//50X*E/N*9X* = *1PE11.4,3X*VOLT CM2*/50X*E/P*9X* = *0PF8.	BOLTZ	911
42,6X*V/CM/TORR*/62X* = *F8.2,6X*KV/CM/ATM*//50X*<U>*9X* = *0PF8.3,	BOLTZ	912
56X*EV*/50X*TE = 2<U>/3K = *F9.0,5X*DEG K*/50X*DIFFUSION D = *	BOLTZ	913



6F9.2,5X*CM2/SEC*/50X*MOBILITY MU = *F8.1,6X*CM2/VOLT/S*/50X*EK =	BOLTZ	914
7D/MU*3X* = *0PF8.3,6X*EV*/50X*VDRIFF =MU*E = #1PE11.4,3X*CM/SEC*/	BOLTZ	915
850X*CONDUCT/NE = *E11.4,3X*CM2/OHM*/50X*CONDUCT*5X* = *E11.4,3X	BOLTZ	916
9*/OHM-CM*/50X*RH0*NE*6X* = *E11.4,3X*OHM/CM2*/50X*J/NE = E*VD = #	BOLTZ	917
\$E11.4,3X*AMP CM/EL*/50X*DISCHARGE*3X* = *E11.4,3X,A10/50X*DEPOSIT	BOLTZ	918
110N = *E11.4,3X,A10/50X*TOTAL POWER = *E11.4,3X,A10//50X*INELAS	BOLTZ	919
2TIC = *E11.4,3X,A10/50X*ELASTIC HEAT = *E11.4,3X,A10/50X*E<U>DN	BOLTZ	920
3E/DT = *E11.4,3X,A10/50X*POWER DISS = *E11.4,3X,A10//50X*NU(MO	BOLTZ	921
4M)*5X* = *E11.4,3X*SEC-1*/50X*NE*10X* = *E11.4,3X*CM-3*/50X*NE/NM	BOLTZ	922
50L*5X* = *E11.4/50X*S(U = 0)/NE = *E11.4,3X*SEC-1*/50X*S(U > 0)/N	BOLTZ	923
6E = *E11.4,3X*SEC-1*/50X*NU(IONIZE) = *E11.4,3X*SEC-1*/50X*NU(A	BOLTZ	924
7TT,REC) = *E11.4,3X*SEC-1*/50X*ONE/DT/NE = *E11.4,3X*SEC-1*/50	BOLTZ	925
8X*ONE/DT*6X* = *E11.4,3X*CM-3/SEC*/48X*42(1H-))	BOLTZ	926
C 110 FORMAT (1H1,9X, *LOG PLOT OF ELECTRON DISTRIBUTION F(U)/F(0) AS A	BOLTZ	927
1FUNCTION OF ELECTRON ENERGY U (EV), WHERE F(0) = *,1PE10.3,* EV*,	BOLTZ	928
22H*.,*(-3/2)*.))	BOLTZ	929
C 120 FORMAT (1H1,17X*LOG PLOT OF RELATIVE ELECTRON DISTRIBUTION FUNCTIO	BOLTZ	930
IN F(U)/FBOLTZ(U,TE), WHERE TE = (2/3)UAVG/KB = *F6.0* DEG K*/	BOLTZ	931
C 130 FORMAT (/13,3X,5A10,A12,A8,3A12,1PE13.3,0PF8.2)	BOLTZ	932
C 140 FORMAT (26X,*U(AVG) = *,F6.3,* EV* TE = *,F6.0,* DEG K, VD = *,	BOLTZ	933
1 1PE8.2* CM/SEC, MU = *E8.2* CM2/VOLT/SEC*/)	BOLTZ	934
C 150 FORMAT (/62X*ELECTRON ENERGY U (EV)*16X*DR. WILLIAM B. LACINA,*A11	BOLTZ	935
1/100X*NORTHROP RESEARCH AND TECHNOLOGY*)	BOLTZ	936
C 160 FORMAT (/1X,135(1H-))	BOLTZ	937
C 170 FORMAT (95X*MOMENTUM TRANSFER = *1PE10.3)	BOLTZ	938
C 180 FORMAT (/1X*F(U) CONVERGED IN*F5.1* SEC* IN*13* ITERATIONS; MAXIMU	BOLTZ	939
\$M RELATIVE CHANGE IN LAST ITERATION <*1PE10.3* POWER BALANCE ACCU	BOLTZ	940
\$RACY = *0PF6.2* %.*)	BOLTZ	941
C 190 FORMAT (1PE10.3)	BOLTZ	942
C 200 FORMAT (15X *NORMALIZED ELECTRON DISTRIBUTION FUNCTION F(U), IN	BOLTZ	943
1 UNITS OF EV*,2H*.,*(-3/2), WITH OUTPUT AT*0PF6.3* EV INTERVALS.*	BOLTZ	944
215X*THE ELECTRON ENERGY RANGE (0,*0PF6.2*) EV WAS SUBDIVIDED INTO	BOLTZ	945
3*14* INTERVALS, GIVING A RESOLUTION = *F5.3* EV.*)	BOLTZ	946
C 210 FORMAT (48X*GAS MIXTURE --//65X*PURE *A3)	BOLTZ	947
C 220 FORMAT (1H(,12,4HX*3,,12,20H(*/*A3),* = *2PF6.2,,12,38H(*/*F6.2),	BOLTZ	948
1* TMOL =*0PF5.0* DEG K*/))	BOLTZ	949
C 230 FORMAT (2X*J*14X*REACTION(J)*21X*N(A)*9X*N(B)*5X*U(J)*4X*DNE/DT/NE	BOLTZ	950
1*2X*<VSIG(A*B)> <VSIG(B*A)> (NET) POWER PERCENT* /13X*A * E(-) *	BOLTZ	951
2B * E(-)*16X*(CM-3)*7X*(CM-3)*4X*(EV)*5X*(SEC-1)*4X*(CM3/SEC)*3X	BOLTZ	952
3*(CM3/SEC)*3X*(A9*) POWER*/1X,135(1H-)/)	BOLTZ	953
C 240 FORMAT (510PF12.3,1PE14.5)/)	BOLTZ	954
C 250 FORMAT (24H(48X,*GAS MIXTURE -- *//,12,5HX,A3,12,20H(*/*A3)* = *,2	BOLTZ	955
	BOLTZ	956
	BOLTZ	957
	BOLTZ	958
	BOLTZ	959
	BOLTZ	960
	BOLTZ	961
	BOLTZ	962
	BOLTZ	963
	BOLTZ	964
	BOLTZ	965
	BOLTZ	966
	BOLTZ	967
	BOLTZ	968
	BOLTZ	969
	BOLTZ	970



	1PF6.2,12,11H(* /PF6.2))	BOLTZ	971
C		BOLTZ	972
	260 FORMAT (55X,*PURE *.A3,* TMOL =*.F5.0,* DEG K*/)	BOLTZ	973
C		BOLTZ	974
	270 FORMAT (48X,4A10//30X*PLASMA KINETICS ANALYSIS WITH SUMMARY OF ELE	BOLTZ	975
	1CTRON PARAMETERS, COLLISION RATES,*/31X*AND POWER BALANCE FOR ALL	BOLTZ	976
	2ELASTIC AND INELASTIC COLLISION PROCESSES INCLUDED*/)	BOLTZ	977
C		BOLTZ	978
	280 FORMAT (/103X, *DR. WILLIAM B. LACINA,*A11/103X*NORTHROP RESEARCH	BOLTZ	979
	1 AND TECHNOLOGY*)	BOLTZ	980
C		BOLTZ	981
	290 FORMAT (2X,132(1H-)/5(7X*U(EV)*7X*F(U)*3X)/2X,132(1H-)/)	BOLTZ	982
C		BOLTZ	983
	300 FORMAT (5H(1H1.,11.4H(/)))	BOLTZ	984
C		BOLTZ	985
	310 FORMAT (82X*E-E POWER TRANSFER DISCREPANCY = *1PE10.3,0PF8.2* %*)	BOLTZ	986
C		BOLTZ	987
	320 FORMAT (34X*FRACTIONAL IONIZATION = NE/NTOT = *1PE10.3*, NE = *	BOLTZ	988
	1 E10.3* CM-3*)	BOLTZ	989
C		BOLTZ	990
	330 FORMAT (31X*E/NTOT = *1PE10.3* VOLT CM2 = *0PF6.3* VOLT/CM/TORR =	BOLTZ	991
	1*F6.3* KVOLT/CM/ATM*)	BOLTZ	992
C		BOLTZ	993
	340 FORMAT (* TOTAL ELECTRICAL = *1PE10.3,A11* 100.00 %,15X*TOTAL PO	BOLTZ	994
	1WER INTO COLLISIONS, HEATING, STORAGE = *1PE10.3,0PF8.2* %*)	BOLTZ	995
C		BOLTZ	996
	350 FORMAT (1H1,48X,4A10/)	BOLTZ	997
C		BOLTZ	998
	360 FORMAT (F7.2)	BOLTZ	999
C		BOLTZ	1000
	370 FORMAT (/1X*DISCHARGE POWER = *1PE10.3,A11,0PF7.2* %,18X*POWER	BOLTZ	1001
	1INTO INELASTIC E-MOLECULE COLLISIONS = *1PE10.3,0PF8.2* %/1X*SOUR	BOLTZ	1002
	2CE DEPOSITION = *1PE10.3,A11,0PF7.2* %,34X*ELASTIC E-MOLECULE HEA	BOLTZ	1003
	3TING = *1PE10.3,0PF8.2* %/22X*-----*12X*-----*14X*D/DT(STORE	BOLTZ	1004
	1D ELECTRON KINETIC ENERGY) = E<U>DNE/DT = *1PE10.3,0PF8.2* %*)	BOLTZ	1005
C		BOLTZ	1006
	380 FORMAT (/48X*CALCULATION PARAMETERS USED:*/48X*MESH =*14*, EMAX =*	BOLTZ	1007
	10PF6.2* EV, DE =*F5.3* EV.*)	BOLTZ	1008
C		BOLTZ	1009
	390 FORMAT (A1,64X*S(U = 0)/NE =*1PE10.3)	BOLTZ	1010
C		BOLTZ	1011
	400 FORMAT (/1X*DISCHARGE POWER = 0.0*46X*POWER INTO INELASTIC E-MOL	BOLTZ	1012
	ECULE COLLISIONS = *1PE10.3,A11/1X*SOURCE DEPOSITION = 0.0*62X*ELA	BOLTZ	1013
	2STIC E-MOLECULE HEATING = *1PE10.3,A11/63X*D/DT(STORED ELECTRON KI	BOLTZ	1014
	3NETIC ENERGY) = E<U>DNE/DT = *1PE10.3,A11)	BOLTZ	1015
C		BOLTZ	1016
	410 FORMAT (* TOTAL ELECTRICAL = 0.0*43X*TOTAL POWER INTO COLLISIONS,	BOLTZ	1017
	1 HEATING, STORAGE = *1PE10.3,A11)	BOLTZ	1018
C		BOLTZ	1019
	420 FORMAT (1H*,124X,F8.2)	BOLTZ	1020
C		BOLTZ	1021
	430 FORMAT (1H*64X,*S(U > 0)/NE =*1PE10.3)	BOLTZ	1022
C		BOLTZ	1023
	500 FORMAT (1H1,20(/)28X,*PROGRAM COMPLETED *,13,* ITERATIONS IN *,	BOLTZ	1024
	1F4.0,* SECONDS WITHOUT SUCCESSFUL CONVERGENCE,*/28X,*REQUESTED LIM	BOLTZ	1025
	2ITS WERE *,13,* ITERATIONS AND *,F4.0,* SECONDS, WITH A CONVERGENC	BOLTZ	1026
	3E CONDI--*/28X,*TION THAT THE CHANGE BETWEEN TWO ITERATIONS BE LESS	BOLTZ	1027

	4 THAN EPS = *.1PE10.3,*,*)	BOLTZ	1028
C		BOLTZ	1029
	600 FORMAT (/28X,*DISTRIBUTION FUNCTION BECAME NEGATIVE AT SOME POINTS	BOLTZ	1030
	1 IN LAST ITERATION*)	BOLTZ	1031
C		BOLTZ	1032
C	-----	BOLTZ	1033
C		BOLTZ	1034
C	CONVERGENCE FAILED FOR LIMITS PRESCRIBED. ERROR MESSAGE --	BOLTZ	1035
C		BOLTZ	1036
	98 WRITE (6,500) ITER, TIME, ITMAX, TMAX, EPS	BOLTZ	1037
	IF (ERROR) WRITE (6,600)	BOLTZ	1038
C		BOLTZ	1039
C	-----	BOLTZ	1040
C		BOLTZ	1041
	99 RETURN	BOLTZ	1042
	END	BOLTZ	1043

	SUBROUTINE SIMEQ (A, M, N, N1, SING)	SIMEQ	2
C		SIMEQ	3
C	.....	SIMEQ	4
C		SIMEQ	5
C	THIS SUBROUTINE WILL SOLVE AN N X N SYSTEM OF SIMULTANEOUS EQUA-	SIMEQ	6
C	TIONS OF THE FORM A(I,J)X(J) = B(I). A IS DIMENSIONED A(M,NN) IN	SIMEQ	7
C	THE MAIN PROGRAM, AND INPUT CONSISTS OF PUTTING THE N X N MATRIX	SIMEQ	8
C	IN THE UPPER LEFT HAND BOX OF A. N1 DIFFERENT B VECTORS CAN BE	SIMEQ	9
C	SPECIFIED AS INPUT STORED IN SUCCESSIVE COLUMNS TO THE RIGHT OF	SIMEQ	10
C	THE N X N MATRIX (N ≤ M, N·N1 ≤ NN). UPON OUTPUT, THE SOLUTION VEC-	SIMEQ	11
C	TORS REPLACE THE INPUT VECTORS B. IF B IS AN N X N UNIT MATRIX ON	SIMEQ	12
C	INPUT, IT WILL CONTAIN A-INVERSE ON OUTPUT. SING = TRUE IF THE	SIMEQ	13
C	MATRIX IS SINGULAR.	SIMEQ	14
C		SIMEQ	15
C	.....	SIMEQ	16
C		SIMEQ	17
	DIMENSION A(M,1)	SIMEQ	18
	LOGICAL SING	SIMEQ	19
	SING = .FALSE.	SIMEQ	20
	NPN1 = N + N1	SIMEQ	21
	DO 9 I = 1,N	SIMEQ	22
	Z1 = 0.	SIMEQ	23
	DO 2 J = 1,N	SIMEQ	24
	X1 = ABS(A(J,I))	SIMEQ	25
	IF (Z1-X1) 1,2,2	SIMEQ	26
1	Z1 = X1	SIMEQ	27
	I1 = J	SIMEQ	28
2	CONTINUE	SIMEQ	29
	IF (Z1) 3,11,3	SIMEQ	30
3	Z = A(I1,I)	SIMEQ	31
	A(I1,I) = A(I,I)	SIMEQ	32
	IP1 = I + 1	SIMEQ	33
	DO 4 L = IP1,NPN1	SIMEQ	34
	X = A(I1,L)	SIMEQ	35
	A(I1,L) = A(I,L)	SIMEQ	36
4	A(I,L) = X/Z	SIMEQ	37
	DO 8 J = 1,N	SIMEQ	38
	IF (J-I) 5,8,5	SIMEQ	39
5	IF (A(J,I).EQ.0.) GO TO 8	SIMEQ	40
	Z = -A(J,I)	SIMEQ	41
	DO 7 L = IP1,NPN1	SIMEQ	42
7	A(J,L) = A(J,L) + Z*A(I,L)	SIMEQ	43
8	CONTINUE	SIMEQ	44
9	CONTINUE	SIMEQ	45
10	RETURN	SIMEQ	46
11	WRITE (6,12)	SIMEQ	47
12	FORMAT (1H1,20(/),40X,*COEFFICIENT MATRIX IS SINGULAR*)	SIMEQ	48
	SING = .TRUE.	SIMEQ	49
	RETURN	SIMEQ	50
	END	SIMEQ	51



	SUBROUTINE GEAR (N, T, Y, SAVE, H, HMIN, HMAX, EPS, MF, YMAX,	GEAR	2
	1 ERROR, KFLAG, JSTART, MAXDER, M, PW)	GEAR	3
C	.....	GEAR	4
C		GEAR	5
C		GEAR	6
C	THIS SUBROUTINE WAS TAKEN FROM THE BOOK, NUMERICAL INITIAL VALUE	GEAR	7
C	PROBLEMS IN ORDINARY DIFFERENTIAL EQUATIONS, BY C. WILLIAM GEAR,	GEAR	8
C	PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, N. J., 1971, PP. 158-166.	GEAR	9
C		GEAR	10
C	THIS SUBROUTINE INTEGRATES A SET OF N ORDINARY DIFFERENTIAL FIRST	GEAR	11
C	ORDER EQUATIONS OVER ONE STEP OF LENGTH H AT EACH CALL. H CAN BE	GEAR	12
C	SPECIFIED BY THE USER AT EACH STEP, BUT IT MAY BE INCREASED OR	GEAR	13
C	DECREASED BY THE PRESENT SUBROUTINE WITHIN THE RANGE HMIN TO HMAX	GEAR	14
C	IN ORDER TO ACHIEVE AS LARGE A STEP AS POSSIBLE WHILE NOT COMMIT-	GEAR	15
C	TING A SINGLE STEP ERROR WHICH IS LARGE THAN EPS IN THE L-2 NORM,	GEAR	16
C	WHERE EACH COMPONENT OF THE ERROR IS DIVIDED BY THE COMPONENTS OF	GEAR	17
C	YMAX. THE PROGRAM REQUIRES THREE SUBROUTINES NAMED:	GEAR	18
C		GEAR	19
C	RATES (N, T, Y, DY)	GEAR	20
C	SIMEQ (PW, M, N, 1, SING)	GEAR	21
C	JACOB (M, T, Y, PW)	GEAR	22
C		GEAR	23
C	THE FIRST, RATES, EVALUATES THE DERIVATIVES OF THE DEPENDENT VARI-	GEAR	24
C	ABLES STORED IN Y(I,1) FOR I = 1 TO N, AND STORES THE DERIVATIVES	GEAR	25
C	IN THE VECTOR DY. THE SECOND, SIMEQ, IS CALLED ONLY IF THE METHOD	GEAR	26
C	FLAG MF IS SET TO 1 OR 2 FOR STIFF METHODS. IT MUST INVERT THE	GEAR	27
C	N X N MATRIX STORED IN THE ARRAY PW(M,M). IF THE INVERSION IS	GEAR	28
C	SUCCESSFUL, SING (SINGULARITY) IS RETURNED FALSE. JACOB IS USED	GEAR	29
C	ONLY IF MF IS 1, AND COMPUTES THE PARTIAL DERIVATIVES OF THE DIF-	GEAR	30
C	FERENTIAL EQUATIONS AS DESCRIBED UNDER THE MF PARAMETER.	GEAR	31
C		GEAR	32
C	THE TEMPORARY STORAGE SPACE IS PROVIDED BY THE CALLER IN THE ARRAY	GEAR	33
C	PW AND THE ARRAY SAVE. THE ARRAY PW IS USED ONLY TO HOLD THE MA-	GEAR	34
C	TRIX OF THE SAME NAME, BUT SAVE IS USED TO HOLD SEVERAL ARRAYS.	GEAR	35
C	THE REGIONS USED ARE:	GEAR	36
C		GEAR	37
C	SAVE(J,1) -- 15J58 AND 1515N IS USED TO SAVE THE VALUES OF	GEAR	38
C	Y IN CASE A STEP HAS TO BE REPEATED.	GEAR	39
C		GEAR	40
C	SAVE(9,1) -- 15 IS USED MAINLY TO HOLD THE CORRECTION TERMS IN	GEAR	41
C	THE CORRECTOR LOOP.	GEAR	42
C		GEAR	43
C	SAVE(10,1) -- 15 IS USED TO SAVE THE VALUES OF THE SUMS OF ALL	GEAR	44
C	OF THE CORRECTION TERMS IN THE PREVIOUS STEP	GEAR	45
C	AFTER THEY HAVE BEEN ACCUMULATED IN THE ARRAY	GEAR	46
C	ERROR IN THE CURRENT STEP. THIS ENABLES THE	GEAR	47
C	BACKWARDS DIFFERENCE OF ERROR TO BE FORMED. IT	GEAR	48
C	IS USED TO ESTIMATE THE STEP SIZE FOR ONE ORDER	GEAR	49
C	HIGHER THAN CURRENT.	GEAR	50
C		GEAR	51
C	SAVE(N1+1,1) -- 15 IS USED TO STORE THE DERIVATIVES WHEN THEY ARE	GEAR	52
C	COMPUTED BY RATES. IT IS ALSO ACCESSED AS	GEAR	53
C	SAVE(N2+1) AS A COMPLETE ARRAY.	GEAR	54
C		GEAR	55
C	SAVE(N5+1,1) -- HOLDS THE DERIVATIVES DURING JACOBIAN EVALUA-	GEAR	56
C	TIONS. IT IS REFERENCED AS SAVE(N6,1) AS A COM-	GEAR	57
C	LETE ARRAY.	GEAR	58



C			GEAR	59
C		INPUT AND OUTPUT PARAMETERS HAVE THE FOLLOWING MEANING --	GEAR	60
C			GEAR	61
C	N	THE NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS. N MAY	GEAR	62
C		BE DECREASED ON LATER CALLS IF THE NUMBER OF ACTIVE EQUA-	GEAR	63
C		TIONS REDUCES, BUT IT MUST NOT BE INCREASED WITHOUT CALL-	GEAR	64
C		ING JSTART = 0.	GEAR	65
C			GEAR	66
C	M	DIMENSION DECLARATOR FOR THE JACOBIAN PW AND THE ARRAY Y	GEAR	67
C		CONTAINING THE DEPENDENT VARIABLES AND THEIR DERIVATIVES.	GEAR	68
C			GEAR	69
C	T	THE INDEPENDENT VARIABLE.	GEAR	70
C			GEAR	71
C	Y	AN M X 8 ARRAY CONTAINING THE DEPENDENT VARIABLES AND	GEAR	72
C		THEIR SCALED DERIVATIVES. Y(I,J+1) CONTAINS THE JTH	GEAR	73
C		DERIVATIVE OF Y(I) SCALED BY H**J/FACTORIAL J, WHERE H IS	GEAR	74
C		THE CURRENT STEP SIZE. ONLY Y(I,1) NEED BE PROVIDED BY	GEAR	75
C		THE CALLING PROGRAM ON THE FIRST ENTRY. IF IT IS DESIRED	GEAR	76
C		TO INTERPOLATE TO NON-MESH POINTS, THESE VALUES CAN BE	GEAR	77
C		USED. IF THE CURRENT STEP SIZE IS H AND THE VALUE AT T+E	GEAR	78
C		IS NEEDED, FORM S = E/H, AND THEN COMPUTE	GEAR	79
C			GEAR	80
C		$Y(I)(T+E) = \sum_{J=0}^{N0} Y(I,J+1) * S ** J$	GEAR	81
C			GEAR	82
C			GEAR	83
C	SAVE	A BLOCK OF AT LEAST 12*N SCRATCH LOCATIONS USED BY THE	GEAR	84
C		SUBROUTINES.	GEAR	85
C			GEAR	86
C			GEAR	87
C	H	THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP. H MAY BE	GEAR	88
C		ADJUSTED UP OR DOWN BY THE PROGRAM IN ORDER TO ACHIEVE AN	GEAR	89
C		ECONOMICAL INTEGRATION. HOWEVER, IF THE H PROVIDED BY THE	GEAR	90
C		USER DOES NOT CAUSE A LARGER ERROR THAN REQUESTED, IT WILL	GEAR	91
C		BE USED. TO SAVE COMPUTER TIME, THE USER IS ADVISED TO	GEAR	92
C		USE A FAIRLY SMALL STEP FOR THE FIRST CALL. IT WILL BE	GEAR	93
C		AUTOMATICALLY INCREASED LATER.	GEAR	94
C			GEAR	95
C	HMIN	THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE INTEGRA-	GEAR	96
C		TION. NOTE THAT ON STARTING THIS MUST BE MUCH SMALLER	GEAR	97
C		THAN THE AVERAGE H EXPECTED SINCE A FIRST ORDER METHOD IS	GEAR	98
C		USED INITIALLY.	GEAR	99
C			GEAR	100
C	HMAX	THE MAXIMUM SIZE TO WHICH THE STEP SIZE WILL BE INCREASED.	GEAR	101
C			GEAR	102
C	EPS	THE ERROR TEST CONSTANT. SINGLE STEP ERROR ESTIMATES DI-	GEAR	103
C		VIDED BY YMAX(I) MUST BE LESS THAN THIS IN THE EUCLIDEAN	GEAR	104
C		NORM. THE STEP AND/OR ORDER IS ADJUSTED TO ACHIEVE THIS.	GEAR	105
C			GEAR	106
C	MF	THE METHOD INDICATOR. THE FOLLOWING ARE AVAILABLE:	GEAR	107
C			GEAR	108
C		0 AN ADAMS PREDICTOR-CORRECTOR IS USED.	GEAR	109
C			GEAR	110
C		1 A MULTISTEP METHOD SUITABLE FOR STIFF SYSTEMS IS	GEAR	111
C		USED. IT WILL ALSO WORK FOR NON-STIFF SYSTEMS.	GEAR	112
C		HOWEVER, THE USER MUST PROVIDE A SUBROUTINE JACOB	GEAR	113
C		WHICH EVALUATES THE PARTIAL DERIVATIVES OF THE	GEAR	114
C		DIFFERENTIAL EQUATIONS WITH RESPECT TO THE Y-S.	GEAR	115

C		THIS IS DONE BY CALL JACOB (M, T, Y, PW(1,N*2)).	GEAR	116
C		THE JACOBIAN PHI(I,J), WHICH REPRESENTS THE PAR-	GEAR	117
C		TIAL OF THE ITH EQUATION WITH RESPECT TO THE JTH	GEAR	118
C		DEPENDENT VARIABLE, IS STORED IN THE UPPER LEFT	GEAR	119
C		N X N CORNER OF THAT PART OF THE ARRAY PW BEGIN-	GEAR	120
C		NING AT COLUMN (N*2).	GEAR	121
C			GEAR	122
C	2	THE SAME AS CASE 1, EXCEPT THAT THIS SUBROUTINE	GEAR	123
C		COMPUTES THE PARTIAL DERIVATIVES BY NUMERICAL DIF-	GEAR	124
C		FERENCING OF THE DERIVATIVES. HENCE, JACOB IS NOT	GEAR	125
C		CALLED.	GEAR	126
C			GEAR	127
C	YMAX	AN ARRAY OF N LOCATIONS WHICH CONTAINS THE MAXIMUM OF EACH	GEAR	128
C		Y SEEN SO FAR, AUTOMATICALLY UPDATED AFTER EACH COMPLETED	GEAR	129
C		STEP (UNLESS THE USER OVERRIDES IT BY CHANGING YMAX BEFORE	GEAR	130
C		A SUBSEQUENT CALL). ALL OF THE COMPONENTS OF YMAX SHOULD	GEAR	131
C		BE INITIALIZED TO 1.0 BEFORE THE FIRST ENTRY. (CF. ALSO	GEAR	132
C		THE DESCRIPTION FOR EPS, GIVEN ABOVE.)	GEAR	133
C			GEAR	134
C	ERROR	AN ARRAY OF N ELEMENTS WHICH CONTAIN THE ESTIMATED ONE	GEAR	135
C		STEP ERROR IN EACH COMPONENT.	GEAR	136
C			GEAR	137
C	KFLAG	A COMPLETION CODE WITH THE FOLLOWING MEANINGS--	GEAR	138
C			GEAR	139
C		+1 THE STEP WAS SUCCESSFUL; T IS ADVANCED TO (T+H)	GEAR	140
C		AND RESULTS OF INTEGRATION FROM T TO (T+H) ARE	GEAR	141
C		RETURNED TO THE CALLING PROGRAM.	GEAR	142
C		-1 THE STEP WAS TAKEN WITH H = HMIN, BUT THE	GEAR	143
C		REQUESTED ERROR WAS NOT ACHIEVED. T IS ADVANCED	GEAR	144
C		TO (T+DT). CALLING PROGRAM MUST APPROVE FAILURE	GEAR	145
C		OF ACCURACY.	GEAR	146
C		-2 THE MAXIMUM ORDER SPECIFIED WAS FOUND TO BE TOO	GEAR	147
C		LARGE.	GEAR	148
C		-3 CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED FOR	GEAR	149
C		H > HMIN. T IS NOT ADVANCED. CALLING PROGRAM	GEAR	150
C		MUST DECREASE H AND HMIN AND TRY AGAIN.	GEAR	151
C		-4 THE REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED	GEAR	152
C		FOR THIS PROBLEM. T IS NOT ADVANCED. ETA MUST	GEAR	153
C		BE INCREASED.	GEAR	154
C			GEAR	155
C	JSTART	INPUT INDICATOR WITH THE FOLLOWING MEANINGS --	GEAR	156
C			GEAR	157
C		-1 REPEAT THE LAST STEP WITH A NEW H	GEAR	158
C		0 PERFORM THE FIRST STEP. THE FIRST STEP MUST BE	GEAR	159
C		DONE WITH THIS VALUE OF JSTART SO THAT THE SUBROU-	GEAR	160
C		TINE CAN INITIALIZE ITSELF.	GEAR	161
C		+1 TAKE A NEW STEP CONTINUING FROM THE LAST.	GEAR	162
C			GEAR	163
C		JSTART IS SET TO NQ, THE CURRENT ORDER OF THE METHOD AT	GEAR	164
C		EXIT. NQ IS ALSO THE ORDER OF THE MAXIMUM DERIVATIVE	GEAR	165
C		AVAILABLE.	GEAR	166
C			GEAR	167
C	MAXDER	THE MAXIMUM DERIVATIVE THAT SHOULD BE USED IN THE METHOD.	GEAR	168
C		SINCE THE ORDER IS EQUAL TO THE HIGHEST DERIVATIVE USED,	GEAR	169
C		THIS RESTRICTS THE ORDER. IT MUST BE LESS THAN 8 FOR	GEAR	170
C		ADAMS AND 7 FOR THE STIFF METHODS.	GEAR	171
C			GEAR	172



C	PW	A BLOCK OF AT LEAST M*(2M+1) SCRATCH LOCATIONS.	GEAR	173
C			GEAR	174
C	.....	.....	GEAR	175
C			GEAR	176
		DIMENSION Y(M,8), YMAX(1), SAVE(10,1), ERROR(1), PW(M,1), A(8),	GEAR	177
		1 PERTST(7,2,3)	GEAR	178
C		LOGICAL SING	GEAR	179
C			GEAR	180
C		THE COEFFICIENTS IN PERTST ARE USED IN SELECTING THE STEP AND	GEAR	181
C		ORDER. THEREFORE, ONLY ABOUT ONE PERCENT ACCURACY IS NEEDED.	GEAR	182
C			GEAR	183
		DATA PERTST / 2.0, 4.5, 7.333, 10.42, 13.7, 17.15, 1.0,	GEAR	184
	1	2.0, 12.0, 24.0, 37.89, 53.33, 70.08, 87.97,	GEAR	185
	2	3.0, 6.0, 9.167, 12.5, 15.98, 1.0, 1.0,	GEAR	186
	3	12.0, 24.0, 37.89, 53.33, 70.08, 87.97, 1.0,	GEAR	187
	4	1.0, 1.0, 0.5, 0.1667, 0.04133, 0.008267, 1.0,	GEAR	188
	5	1.0, 1.0, 2.0, 1.0, 0.3157, 0.07407, 0.0139 /	GEAR	189
C			GEAR	190
		DATA A(2) / -1.0 /	GEAR	191
		IRET = 1	GEAR	192
		KFLAG = 1	GEAR	193
		METHOD = MF+1	GEAR	194
		IF (JSTART.LE.0) GO TO 140	GEAR	195
C			GEAR	196
C		BEGIN BY SAVING INFORMATION FOR POSSIBLE RESTARTS AND CHANGING H	GEAR	197
C		BY THE FACTOR R IF THE CALLER HAS CHANGED H. ALL VARIABLES DEPEND	GEAR	198
C		ENT ON H MUST ALSO BE CHANGED. E IS A COMPARISON FOR ERRORS OF	GEAR	199
C		THE CURRENT ORDER, NQ. EUP IS TO TEST FOR INCREASING THE ORDER.	GEAR	200
C		EDWN FOR DECREASING THE ORDER. HNEW IS THE STEP SIZE THAT WAS	GEAR	201
C		USED ON THE LAST CALL.	GEAR	202
C			GEAR	203
			GEAR	204
	100	DO 110 I = 1,N	GEAR	205
		DO 110 J = 1,K	GEAR	206
	110	SAVE(J,I) = Y(I,J)	GEAR	207
		HOLD = HNEW	GEAR	208
		IF (H.EQ.HOLD) GO TO 130	GEAR	209
	120	RACUM = H/HOLD	GEAR	210
		IRET1 = 1	GEAR	211
		GO TO 750	GEAR	212
	130	NWOLD = NQ	GEAR	213
		TOLD = T	GEAR	214
		RACUM = 1.0	GEAR	215
		IF (JSTART.GT.0) GO TO 250	GEAR	216
		GO TO 170	GEAR	217
	140	IF (JSTART.EQ.-1) GO TO 160	GEAR	218
C			GEAR	219
C		ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL DERIVA-	GEAR	220
C		TIVES ARE CALCULATED.	GEAR	221
C			GEAR	222
		NQ = 1	GEAR	223
		N1 = N+10	GEAR	224
		N2 = N1 + 1	GEAR	225
		N5 = N1 + N	GEAR	226
		N6 = N5 + 1	GEAR	227
C			GEAR	228
C			GEAR	229



C	-----	GEAR	230
	CALL DNDT (N, T, Y, SAVE(N2,1))	GEAR	231
C	-----	GEAR	232
C		GEAR	233
	DO 150 I = 1,N	GEAR	234
150	Y(I,2) = SAVE(N1,I,1)*H	GEAR	235
	HNEW = H	GEAR	236
	K = 2	GEAR	237
	GO TO 100	GEAR	238
C	REPEAT LAST STEP BY RESTORING SAVED INFORMATION --	GEAR	239
160	IF (NQ,EQ,NQOLD) JSTART = 1	GEAR	240
	T = TOLD	GEAR	241
	NQ = NQOLD	GEAR	242
	K = NQ*1	GEAR	243
	GO TO 120	GEAR	244
C		GEAR	245
C	SET THE COEFFICIENTS THAT DETERMINE THE ORDER AND THE METHOD TYPE.	GEAR	246
C	CHECK FOR EXCESSIVE ORDER. THE LAST TWO STATEMENTS OF THIS SEC-	GEAR	247
C	TION SET IWEAL > 0 IF PW IS TO BE REEVALUATED BECAUSE OF THE OR-	GEAR	248
C	DER CHANGE, AND THEN REPEAT THE INTEGRATION STEP IF IT HAS NOT YET	GEAR	249
C	BEEN DONE (IRET = 1) OR SKIP TO A FINAL SCALING BEFORE EXIT IF IT	GEAR	250
C	HAS BEEN COMPLETED (IRET = 2).	GEAR	251
C		GEAR	252
170	IF (MF,EQ,0) GO TO 180	GEAR	253
	IF (NQ,GT,6) GO TO 190	GEAR	254
	GO TO (221,222,223,224,225,226), NQ	GEAR	255
180	IF (NQ,GT,7) GO TO 190	GEAR	256
	GO TO (211,212,213,214,215,216,217), NQ	GEAR	257
190	KFLAG = -2	GEAR	258
	RETURN	GEAR	259
C		GEAR	260
211	A(1) = -1.0	GEAR	261
	GO TO 230	GEAR	262
212	A(1) = A(3) = -0.5	GEAR	263
	GO TO 230	GEAR	264
213	A(1) = -0.416666666666667	GEAR	265
C	A(1) = -5./12.	GEAR	266
	A(3) = -0.75	GEAR	267
	A(4) = -0.166666666666667	GEAR	268
C	A(4) = -1./6.	GEAR	269
	GO TO 230	GEAR	270
214	A(1) = -.375	GEAR	271
	A(3) = -0.916666666666667	GEAR	272
C	A(3) = -11./12.	GEAR	273
	A(4) = -0.333333333333333	GEAR	274
C	A(4) = -1./3.	GEAR	275
	A(5) = -0.416666666666667E-01	GEAR	276
C	A(5) = -1./24.	GEAR	277
	GO TO 230	GEAR	278
215	A(1) = -0.348611111111111	GEAR	279
C	A(1) = -251./720.	GEAR	280
	A(3) = -1.041666666666667	GEAR	281
C	A(3) = -25./24.	GEAR	282
	A(4) = -0.486111111111111	GEAR	283
C	A(4) = -35./72.	GEAR	284
	A(5) = -0.104166666666667	GEAR	285
C	A(5) = -5./48.	GEAR	286

A(6) = -0.83333333333333E-02  
 C A(6) = -1./120.  
 GO TO 230  
 216 A(1) = -0.32986111111111  
 C A(1) = -95./288.  
 A(3) = -1.1416666666667  
 C A(3) = -137./120.  
 A(4) = -0.625  
 C A(4) = -5./8.  
 A(5) = -0.17708333333333  
 C A(5) = -17./96.  
 A(6) = -0.025  
 C A(6) = -1./40.  
 A(7) = -0.13888888888889E-02  
 C A(7) = -1./720.  
 GO TO 230  
 217 A(1) = -0.31559193121693  
 C A(1) = -19087./60480.  
 A(3) = -1.225  
 C A(3) = -49./40.  
 A(4) = -0.75185185185185  
 C A(4) = -203./270.  
 A(5) = -0.25520833333333  
 C A(5) = -49./192.  
 A(6) = -0.48611111111111E-01  
 C A(6) = -7./144.  
 A(7) = -0.48611111111111E-02  
 C A(7) = -7./1440.  
 A(8) = -0.19841269841270E-03  
 C A(8) = -1./5040.  
 GO TO 230  
 221 A(1) = -1.  
 GO TO 230  
 222 A(1) = -0.66666666666667  
 C A(1) = -2./3.  
 A(3) = -0.33333333333333  
 C A(3) = -1./3.  
 GO TO 230  
 223 A(1) = -0.54545454545454  
 A(3) = A(1)  
 C A(1) = A(3) = -6./11.  
 A(4) = -0.90909090909091E-01  
 C A(4) = -1./11.  
 GO TO 230  
 224 A(1) = -0.48  
 A(3) = -0.7  
 A(4) = -0.2  
 A(5) = -0.02  
 GO TO 230  
 225 A(1) = -0.43795620437956  
 C A(1) = -120./274.  
 A(3) = -0.82116788321168  
 C A(3) = -225./274.  
 A(4) = -0.31021897810219  
 C A(4) = -85./274.  
 A(5) = -0.54744525547445E-01  
 C A(5) = -15./274.

GEAR 287  
 GEAR 288  
 GEAR 289  
 GEAR 290  
 GEAR 291  
 GEAR 292  
 GEAR 293  
 GEAR 294  
 GEAR 295  
 GEAR 296  
 GEAR 297  
 GEAR 298  
 GEAR 299  
 GEAR 300  
 GEAR 301  
 GEAR 302  
 GEAR 303  
 GEAR 304  
 GEAR 305  
 GEAR 306  
 GEAR 307  
 GEAR 308  
 GEAR 309  
 GEAR 310  
 GEAR 311  
 GEAR 312  
 GEAR 313  
 GEAR 314  
 GEAR 315  
 GEAR 316  
 GEAR 317  
 GEAR 318  
 GEAR 319  
 GEAR 320  
 GEAR 321  
 GEAR 322  
 GEAR 323  
 GEAR 324  
 GEAR 325  
 GEAR 326  
 GEAR 327  
 GEAR 328  
 GEAR 329  
 GEAR 330  
 GEAR 331  
 GEAR 332  
 GEAR 333  
 GEAR 334  
 GEAR 335  
 GEAR 336  
 GEAR 337  
 GEAR 338  
 GEAR 339  
 GEAR 340  
 GEAR 341  
 GEAR 342  
 GEAR 343

	A(6) = -0.36496350364964E-02	GEAR	344
C	A(6) = -1./274.	GEAR	345
	GO TO 230	GEAR	346
226	A(1) = -0.40816326530612	GEAR	347
C	A(1) = -180./441.	GEAR	348
	A(3) = -0.92063492063492	GEAR	349
C	A(3) = -58./63.	GEAR	350
	A(4) = -0.41666666666667	GEAR	351
C	A(4) = -15./36.	GEAR	352
	A(5) = -0.99206349206349E-01	GEAR	353
C	A(5) = -25./252.	GEAR	354
	A(6) = -0.11904761904762E-01	GEAR	355
C	A(6) = -3./252.	GEAR	356
	A(7) = -0.56689342403628E-03	GEAR	357
C	A(7) = -1./1764.	GEAR	358
		GEAR	359
230	K = NQ+1	GEAR	360
	IDOUB = K	GEAR	361
	MTYP = (4-MF)/2	GEAR	362
C	MTYP = 1 (STIFF, MF = 1 OR 2), MTYP = 2 (ADAMS-MOULTON).	GEAR	363
	ENQ2 = 0.5/(NQ+1)	GEAR	364
	ENQ3 = 0.5/(NQ+2)	GEAR	365
	ENQ1 = 0.5/NQ	GEAR	366
	PEPSH = EPS	GEAR	367
	EUP = (PERTST(NQ,MTYP,2)*PEPSH)**2	GEAR	368
	E = (PERTST(NQ,MTYP,1)*PEPSH)**2	GEAR	369
	EDWN = (PERTST(NQ,MTYP,3)*PEPSH)**2	GEAR	370
	IF (EDWN.EQ.0) GO TO 780	GEAR	371
	BND = EPS*ENQ3/N	GEAR	372
	IWEVAL = MF	GEAR	373
	GO TO (250+680), IRET	GEAR	374
C		GEAR	375
C	THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY MULTI-	GEAR	376
C	PLYING THE SAVED INFORMATION BY THE PASCAL TRIANGLE MATRIX.	GEAR	377
C		GEAR	378
250	T = T+H	GEAR	379
	DO 260 J = 2,K	GEAR	380
	DO 260 J1 = J,K	GEAR	381
	J2 = K-J1+J-1	GEAR	382
	DO 260 I = 1,N	GEAR	383
260	Y(I,J2) = Y(I,J2) + Y(I,J2+1)	GEAR	384
C		GEAR	385
C	UP TO 3 CORRECTOR ITERATIONS ARE TAKEN. CONVERGENCE IS TESTED BY	GEAR	386
C	REQUIRING CHANGES TO BE LESS THAN BND WHICH IS DEPENDENT ON THE	GEAR	387
C	ERROR TEST CONSTANT. THE SUM OF THE CORRECTIONS IS ACCUMULATED IN	GEAR	388
C	THE ARRAY ERROR(I). IT IS EQUAL TO THE KTH DERIVATIVE OF Y MULTI-	GEAR	389
C	PLIED BY H**K/(FACTORIAL(K-1)*A(K)), AND IS THEREFORE PROPOR-	GEAR	390
C	TIONAL TO THE ACTUAL ERRORS TO THE LOWEST POWER OF H PRESENT (H**K	GEAR	391
C		GEAR	392
	DO 270 I = 1,N	GEAR	393
270	ERROR(I) = 0.	GEAR	394
C		GEAR	395
	DO 430 L = 1,3	GEAR	396
C		GEAR	397
C	-----	GEAR	398
	CALL DNDT (N, T, Y, SAVE(N2,1))	GEAR	399
C	-----	GEAR	400



C	GO TO (280,300,320), METHOD	GEAR	401
C		GEAR	402
C	THIS SECTION IS ENCOUNTERED ONLY FOR MF = 0 --	GEAR	403
C		GEAR	404
	280 DO 290 I = 1,N	GEAR	405
	290 SAVE(9,I) = Y(I,2) - SAVE(N1+I,1)*H	GEAR	406
	GO TO 410	GEAR	407
C		GEAR	408
C	-----	GEAR	409
C	EVALUATE THE JACOBIAN AND PLACE IT IN AN N X N BOX IN THE UPPER	GEAR	410
C	LEFT HAND CORNER OF THAT PART OF PW BEGINNING WITH COLUMN (N+2).	GEAR	411
C	-----	GEAR	412
C		GEAR	413
C	EVALUATE THE JACOBIAN BY A CALL TO AN EXTERNAL SUBROUTINE (THIS	GEAR	414
C	SECTION IS ENCOUNTERED ONLY IF MF = 1) --	GEAR	415
C		GEAR	416
C	300 IF (IWEVAL.LT.1) GO TO 380	GEAR	417
C		GEAR	418
C	-----	GEAR	419
C	CALL JACOB (M, T, Y, PW(1,N+2))	GEAR	420
C	-----	GEAR	421
C		GEAR	422
	R = A(1)*H	GEAR	423
	DO 310 I = 1,N	GEAR	424
	DO 310 J = 1,N	GEAR	425
	JPN1 = J+N+1	GEAR	426
	310 PW(I,JPN1) = PW(I,JPN1)*R	GEAR	427
	GO TO 360	GEAR	428
C		GEAR	429
C	.....	GEAR	430
C		GEAR	431
C	EVALUATE THE JACOBIAN INTO PW BY NUMERICAL DIFFERENCING. R IS THE	GEAR	432
C	CHANGE MADE TO THE ELEMENT OF Y. IT IS EPS RELATIVE TO Y WITH A	GEAR	433
C	MINIMUM OF EPS**2. THIS SECTION IS ENCOUNTERED ONLY IF MF = 2 --	GEAR	434
C		GEAR	435
	320 IF (IWEVAL.LT.1) GO TO 380	GEAR	436
	DO 330 I = 1,N	GEAR	437
	330 SAVE(9,I) = Y(I,1)	GEAR	438
	DO 350 J = 1,N	GEAR	439
	R = EPS*AMAX1(EPS,ABS(SAVE(9,J)))	GEAR	440
	Y(J,1) = Y(J,1) + R	GEAR	441
	D = A(1)*H/R	GEAR	442
C		GEAR	443
C	-----	GEAR	444
C	CALL DNDDT (N, T, Y, SAVE(N6+1))	GEAR	445
C	-----	GEAR	446
C		GEAR	447
	JPN1 = J+N+1	GEAR	448
	DO 340 I = 1,N	GEAR	449
	340 PW(I,JPN1) = (SAVE(N5+I,1) - SAVE(N1+I,1))*D	GEAR	450
	350 Y(J,1) = SAVE(9,J)	GEAR	451
C		GEAR	452
C	.....	GEAR	453
C		GEAR	454
C	IF THERE HAS BEEN A CHANGE OF ORDER OR THERE HAS BEEN TROUBLE WITH	GEAR	455
C	CONVERGENCE, PW IS REEVALUATED PRIOR TO STARTING THE CORRECTOR	GEAR	456
C		GEAR	457

AD-A066 156

NORTHROP RESEARCH AND TECHNOLOGY CENTER PALOS VERDES --ETC F/G 20/5  
THEORETICAL MODELING OF MOLECULAR AND ELECTRON KINETIC PROCESSES--ETC(U)  
JAN 79 W B LACINA

N00014-78-C-0499

NL

UNCLASSIFIED

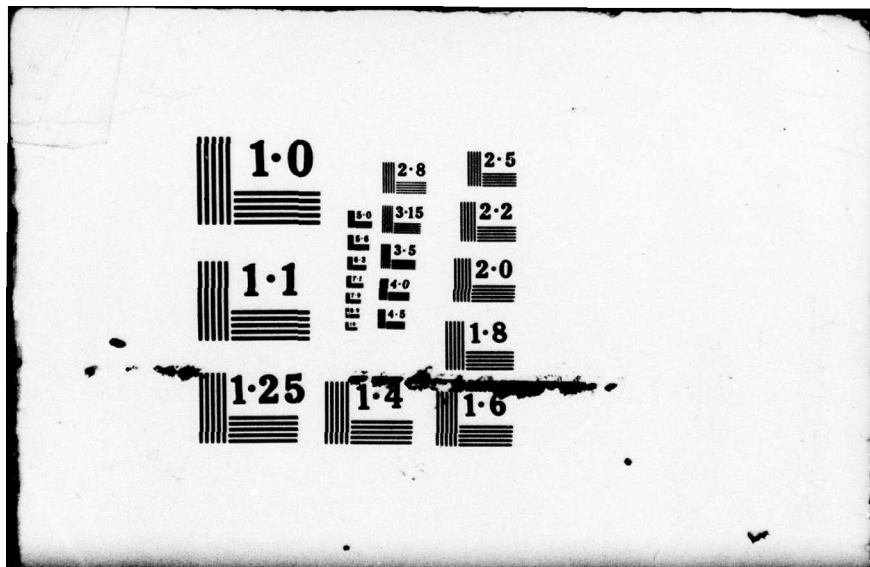
NRTC-79-TR-VOL-2

2 OF 2  
ADA  
066156



END  
DATE  
FILMED

5 79  
DOC





C	ITERATION IN THE CASE OF STIFF METHODS. IWEVAL IS THEN SET TO -1	GEAR	458
C	AS AN INDICATOR THAT IT HAS BEEN DONE.	GEAR	459
C		GEAR	460
C	360 IWEVAL = -1	GEAR	461
C		GEAR	462
C	ADD THE IDENTITY MATRIX TO THE JACOBIAN. THIS SECTION IS ENCOUN-	GEAR	463
C	TERED ONLY FOR MF = 1 OR 2 --	GEAR	464
C		GEAR	465
C	DO 370 I = 1,N	GEAR	466
C	IPN1 = I+N+1	GEAR	467
C	370 PW(I,IPN1) = 1.0 + PW(I,IPN1)	GEAR	468
C		GEAR	469
C	FETCH (1 + PHI), SHIFT IT TO THE LEFT, CALCULATE RHS VECTOR --	GEAR	470
C	380 DO 390 I = 1,N	GEAR	471
C	PW(I,N+1) = SAVE(N+I,1) = Y(I,2) - SAVE(N+I,1)*H	GEAR	472
C	DO 390 J = 1,N	GEAR	473
C	JPN1 = J+N+1	GEAR	474
C	390 PW(I,J) = PW(I,JPN1)	GEAR	475
C		GEAR	476
C	-----	GEAR	477
C	CALL SIMEU (PW, M, N, 1, SING)	GEAR	478
C	-----	GEAR	479
C		GEAR	480
C	IF (SING) GO TO 440	GEAR	481
C		GEAR	482
C	DO 400 I = 1,N	GEAR	483
C	400 SAVE(9,I) = PW(I,N+1)	GEAR	484
C		GEAR	485
C	.....	GEAR	486
C		GEAR	487
C	CORRECT AND SEE IF ALL OF THE CHANGES ARE LESS THAN BND RELATIVE	GEAR	488
C	TO YMAX. IF SO, THE CORRECTOR IS SAID TO HAVE CONVERGED.	GEAR	489
C		GEAR	490
C	410 NT = N	GEAR	491
C	DO 420 I = 1,N	GEAR	492
C	Y(I,1) = Y(I,1) + A(I)*SAVE(9,I)	GEAR	493
C	Y(I,2) = Y(I,2) - SAVE(9,I)	GEAR	494
C	ERROR(I) = ERROR(I) + SAVE(9,I)	GEAR	495
C	IF (ABS(SAVE(9,I)).LE.(BND*YMAX(I))) NT = NT-1	GEAR	496
C	420 CONTINUE	GEAR	497
C	IF (NT.LE.0) GO TO 490	GEAR	498
C	430 CONTINUE	GEAR	499
C		GEAR	500
C	.....	GEAR	501
C		GEAR	502
C	THE CORRECTOR ITERATION FAILED TO CONVERGE IN THREE TRIES. VARI-	GEAR	503
C	OUS POSSIBILITIES ARE CHECKED FOR. IF H IS ALREADY HMIN AND THIS	GEAR	504
C	IS EITHER ADAMS METHOD OR THE STIFF METHOD IN WHICH THE MATRIX PW	GEAR	505
C	HAS ALREADY BEEN REEVALUATED, A NO-CONVERGENCE EXIT IS TAKEN.	GEAR	506
C	OTHERWISE, THE MATRIX PW IS REEVALUATED AND/OR THE STEP IS RE-	GEAR	507
C	DUCE TO TRY AND GET CONVERGENCE.	GEAR	508
C		GEAR	509
C	.....	GEAR	510
C		GEAR	511
C	440 T = TOLD	GEAR	512
C	IF ((H.LE.(HMIN*.00001)).AND.((IWEVAL-MTYP).LT.-1)) GO TO 460	GEAR	513
C	IF ((MF.EQ.0).OR.(IWEVAL.NE.0)) RACUM = RACUM*0.25	GEAR	514

IWEVAL = MF	GEAR	515
IREF1 = 2	GEAR	516
GO TO 750	GEAR	517
460 KFLAG = -3	GEAR	518
470 DO 480 I = 1,N	GEAR	519
DO 480 J = 1,K	GEAR	520
480 Y(I,J) = SAVE(J,I)	GEAR	521
H = HOLD	GEAR	522
NH = NQOLD	GEAR	523
JSTART = NH	GEAR	524
RETURN	GEAR	525
C	GEAR	526
C	GEAR	527
C	GEAR	528
C	GEAR	529
C	GEAR	530
C	GEAR	531
C	GEAR	532
C	GEAR	533
C	GEAR	534
C	GEAR	535
C	GEAR	536
C	GEAR	537
C	GEAR	538
C	GEAR	539
C	GEAR	540
C	GEAR	541
C	GEAR	542
C	GEAR	543
C	GEAR	544
C	GEAR	545
490 D = 0.	GEAR	546
DO 500 I = 1,N	GEAR	547
500 D = D + (ERROR(I)/YMAX(I))*2	GEAR	548
IWEVAL = 0	GEAR	549
IF (D.GT.E) GO TO 540	GEAR	550
IF (K.LT.J) GO TO 520	GEAR	551
C	GEAR	552
C	GEAR	553
C	GEAR	554
DO 510 J = 3,K	GEAR	555
DO 510 I = 1,N	GEAR	556
510 Y(I,J) = Y(I,J) + A(J)*ERROR(I)	GEAR	557
520 KFLAG = 1	GEAR	558
MNEW = H	GEAR	559
IF (IDOUB.LE.1) GO TO 550	GEAR	560
IDOUB = IDOUB-1	GEAR	561
IF (IDOUB.GT.1) GO TO 700	GEAR	562
DO 530 I = 1,N	GEAR	563
530 SAVE(10,I) = ERROR(I)	GEAR	564
GO TO 700	GEAR	565
C	GEAR	566
C	GEAR	567
C	GEAR	568
C	GEAR	569
C	GEAR	570
C	GEAR	571



540	KFLAG = KFLAG-2	GEAR	572
	IF (H.LE.(HMIN*1.00001)) GO TO 740	GEAR	573
	T = TOLD	GEAR	574
	IF (KFLAG.LE.-5) GO TO 720	GEAR	575
C		GEAR	576
C	PR1, PR2, AND PR3 WILL CONTAIN THE AMOUNTS BY WHICH THE STEP SIZE	GEAR	577
C	SHOULD BE DIVIDED AT ORDER ONE LOWER, AT THIS ORDER, AND AT ORDER	GEAR	578
C	ONE HIGHER, RESPECTIVELY.	GEAR	579
C		GEAR	580
550	PR2 = (D/E)**ENQ2*1.2	GEAR	581
	PR3 = 1.E 20	GEAR	582
	IF ((NQ.GE.MAXDER).OR.(KFLAG.LE.-1)) GO TO 570	GEAR	583
	D = 0.	GEAR	584
	DO 560 I = 1,N	GEAR	585
560	D = D + ((ERROR(I) - SAVE(I0,I))/YMAX(I))**2	GEAR	586
	PR3 = (D/EUP)**ENQ3*1.4	GEAR	587
570	PR1 = 1.E 20	GEAR	588
	IF (NQ.LE.1) GO TO 590	GEAR	589
	D = 0.	GEAR	590
	DO 580 I = 1,N	GEAR	591
580	D = D + (Y(I,K)/YMAX(I))**2	GEAR	592
	PR1 = (D/EDWN)**ENQ1*1.3	GEAR	593
590	CONTINUE	GEAR	594
	IF (PR2.LE.PR3) GO TO 650	GEAR	595
	IF (PR3.LT.PR1) GO TO 660	GEAR	596
600	R = 1.0/AMAX1(PR1,1.E-04)	GEAR	597
	NEWQ = NQ-1	GEAR	598
610	IDOUB = 10	GEAR	599
	IF ((KFLAG.EQ.1).AND.(R.LT.1.1)) GO TO 700	GEAR	600
	IF (NEWQ.LE.NQ) GO TO 630	GEAR	601
C	COMPUTE ONE ADDITIONAL SCALED DERIVATIVE IF ORDER IS INCREASED.	GEAR	602
	DO 620 I = 1,N	GEAR	603
620	Y(I,NEWQ+1) = ERROR(I)*A(K)/K	GEAR	604
630	K = NEWQ+1	GEAR	605
	IF (KFLAG.EQ.1) GO TO 670	GEAR	606
	RACUM = RACUM*R	GEAR	607
	IRET1 = 3	GEAR	608
	GO TO 750	GEAR	609
640	IF (NEWQ.EQ.NQ) GO TO 250	GEAR	610
	NQ = NEWQ	GEAR	611
	GO TO 170	GEAR	612
650	IF (PR2.GT.PR1) GO TO 600	GEAR	613
	NEWQ = NQ	GEAR	614
	R = 1.0/AMAX1(PR2,1.E-04)	GEAR	615
	GO TO 610	GEAR	616
660	R = 1.0/AMAX1(PR3,1.E-04)	GEAR	617
	NEWQ = NQ+1	GEAR	618
	GO TO 610	GEAR	619
670	IRET = 2	GEAR	620
	R = AMIN1(R,HMAX/ABS(H))	GEAR	621
	H = H*R	GEAR	622
	HNEW = H	GEAR	623
	IF (NQ.EQ.NEWQ) GO TO 680	GEAR	624
	NQ = NEWQ	GEAR	625
	GO TO 170	GEAR	626
680	R1 = 1.	GEAR	627
	DO 690 J = 2,K	GEAR	628



	R1 = R1*R	GEAR	629
	DO 690 I = 1,N	GEAR	630
690	Y(I,J) = Y(I,J)*R1	GEAR	631
	IDOUB = K	GEAR	632
700	DO 710 I = 1,N	GEAR	633
710	YMAX(I) = AMAX1(YMAX(I), ABS(Y(I,1)))	GEAR	634
	JSTART = NQ	GEAR	635
	RETURN	GEAR	636
720	IF (NQ.EQ.1) GO TO 780	GEAR	637
C		GEAR	638
C	-----	GEAR	639
	CALL DNDT (N, T, Y, SAVE(N2,1))	GEAR	640
C	-----	GEAR	641
C		GEAR	642
	R = H/HOLD	GEAR	643
	DO 730 I = 1,N	GEAR	644
	Y(I,1) = SAVE(1,I)	GEAR	645
	SAVE(2,I) = HOLD*SAVE(N1+1,I)	GEAR	646
730	Y(I,2) = SAVE(2,I)*R	GEAR	647
	NQ = 1	GEAR	648
	KFLAG = 1	GEAR	649
	GO TO 170	GEAR	650
740	KFLAG = -1	GEAR	651
	HNEW = H	GEAR	652
	JSTART = NQ	GEAR	653
	RETURN	GEAR	654
C		GEAR	655
C	THIS SECTION SCALES ALL VARIABLES CONNECTED WITH H AND RETURNS	GEAR	656
C	TO THE ENTERING SECTION.	GEAR	657
		GEAR	658
750	RACUM = AMAX1(ABS(HMIN/HOLD),RACUM)	GEAR	659
	RACUM = AMIN1(RACUM,ABS(HMAX/HOLD))	GEAR	660
	R1 = 1.0	GEAR	661
	DO 760 J = 2,K	GEAR	662
	R1 = R1*RACUM	GEAR	663
	DO 760 I = 1,N	GEAR	664
760	Y(I,J) = SAVE(J,I)*R1	GEAR	665
	H = HOLD*RACUM	GEAR	666
	DO 770 I = 1,N	GEAR	667
770	Y(I,1) = SAVE(1,I)	GEAR	668
	IDOUB = K	GEAR	669
	GO TO (130,250,640), IRET1	GEAR	670
780	KFLAG = -4	GEAR	671
	GO TO 470	GEAR	672
	RETURN	GEAR	673
	END	GEAR	674

	SUBROUTINE PLOT (MM, MP, MULT, Y, Y0, DY, X, X0, DX, SCALEX,	PLOT	2
	1 SCALEY, SAME, CLEAR, CENTER, NAME, NP, IP)	PLOT	3
C		PLOT	4
C	.....	PLOT	5
C		PLOT	6
C	THIS PROGRAM GENERATES LINEAR, SEMILOG, OR LOGLOG PLOTS FOR UP TO 10	PLOT	7
C	VECTORS Y(I,J), J = 1,2,...,NP, PROVIDED BY AN ARRAY DIMENSIONED	PLOT	8
C	Y(MM,...) IN THE CALLING PROGRAM. EACH VECTOR IS CONSIDERED TO BE A	PLOT	9
C	FUNCTION OF AN INDEPENDENT VARIABLE X(I), DEFINED BELOW. ALL PLOTS	PLOT	10
C	ARE GENERATED FOR EACH VECTOR BY SAMPLING MP POINTS, TAKEN WITH A RE-	PLOT	11
C	PETITION INDEX MULT (I.E., I = 1, (1*MULT), ..., (1 + (MP-1)*MULT)).	PLOT	12
C	HORIZONTALLY, THE PLOTTING RESOLUTION CONSISTS OF A MAXIMUM OF 50	PLOT	13
C	SUBINTERVALS, CORRESPONDING TO 51 POINTS. IF MP ≤ 51, MP POINTS ARE	PLOT	14
C	PLOTTED. THERE ARE OPTIONS (AS WELL AS PROGRAM DEFAULTS) FOR SPECI-	PLOT	15
C	FYING INITIAL VALUES AND TICK MARK INTERVALS EITHER BY DIRECT INPUT,	PLOT	16
C	OR BY INTERNAL AUTOMATIC SCALING. AUTOMATIC SCALING, WHICH IS ACCOM-	PLOT	17
C	PLISHED BY SUBROUTINES *AXIS* AND *INTERP*, GENERATES CONVENIENT IN-	PLOT	18
C	TEGER VALUES FOR INITIAL VALUES AND TICK MARK INTERVALS. IF SCALEX	PLOT	19
C	= TRUE IS SPECIFIED, AUTOMATIC SCALING OF THE HORIZONTAL AXIS OCCURS.	PLOT	20
C	IF MP > 51, THE PROGRAM DEFAULTS TO AUTOMATIC SCALING FOR THE X-AXIS,	PLOT	21
C	AND INTERNALLY GENERATES, BY INTERPOLATION, 51 POINTS TO BE PLOTTED.	PLOT	22
C	IT IS ASSUMED THAT THE VECTOR X IS DEFINED BY X(I) = X0 + (I-1)*DX.	PLOT	23
C	IN WHICH CASE, THE X-ORIGIN IS X0 AND THE TICK MARK INTERVAL IS 5*DX.	PLOT	24
C	HOWEVER, THERE ARE TWO EXCEPTIONS IF DX = 0: 1) FOR AUTOMATIC X-AXIS	PLOT	25
C	SCALING, THE INPUT VECTOR X(I) IS USED, AND 2) IF MP ≤ 51, IT IS AS-	PLOT	26
C	SUMED THAT THE INDEPENDENT VARIABLE IS JUST THE INTEGER I. (THUS, IF	PLOT	27
C	DATA IS DEFINED OVER A NONUNIFORM GRID X(I), SCALEX = TRUE AND DX = 0	PLOT	28
C	SHOULD BE SPECIFIED BY THE CALLING PROGRAM.) IF SCALEY = TRUE, AUTO-	PLOT	29
C	MATIC SCALING OF THE VERTICAL AXIS OCCURS FOR THE PLOT(S). IF SCALEY	PLOT	30
C	= FALSE, THE ORIGIN(S) AND TICK MARK INTERVAL(S) FOR THE VERTICAL	PLOT	31
C	AXIS (DIVIDED INTO TEN TICKS) ARE TAKEN TO BE THOSE SPECIFIED BY THE	PLOT	32
C	VECTORS Y0(2) AND DY(2), UNLESS DY(1) OR DY(2) = 0, IN WHICH CASE	PLOT	33
C	ONE, BOTH, OR ALL VECTORS ARE AUTOMATICALLY SCALED BY DEFAULT. IF	PLOT	34
C	SAME = TRUE, PLOT(S) ARE SCALED TOGETHER USING SPECIFIED VALUES Y0(1)	PLOT	35
C	AND DY(1) IF DY(1) ≠ 0, OR AUTOMATIC SCALE VALUES IF DY(1) = 0. IN	PLOT	36
C	THE CASE THAT SAME = TRUE, A SINGLE (COMMON) VERTICAL SCALE FOR THE	PLOT	37
C	PLOT(S) WILL APPEAR AT THE LEFT. IF SAME = FALSE AND TWO PLOTS (NP =	PLOT	38
C	2) ARE REQUESTED, DIFFERENT LEFT AND RIGHT HAND SCALING WILL APPEAR.	PLOT	39
C	PLOTS WILL BE SMOOTHED BY INTERPOLATION, AND WILL BE CENTERED ON THE	PLOT	40
C	PAGE (IF REQUESTED).	PLOT	41
C		PLOT	42
C	MISCELLANEOUS CONDITIONS --	PLOT	43
C		PLOT	44
C	1) IF THE DATA HAS A TOTAL RANGE THAT IS LESS THAN 1.E-04 TIMES ITS	PLOT	45
C	AVERAGE VALUE, THE SMALL (AC) VARIATIONS WILL BE PLOTTED WITH A	PLOT	46
C	(DC) BASELINE VALUE SPECIFIED.	PLOT	47
C		PLOT	48
C	2) PROGRAM DEFAULTS TO FIRST 10 VECTORS IF NP > 10.	PLOT	49
C		PLOT	50
C	3) PROGRAM DEFAULTS TO PLOT OF FIRST VECTOR ONLY IF HISTOGRAM IS	PLOT	51
C	REQUESTED (CLEAR = FALSE).	PLOT	52
C		PLOT	53
C	4) IF NP = 2 AND SAME = TRUE, PROGRAM DEFAULTS TO SAME = FALSE IF	PLOT	54
C	THE TOTAL RANGE OF THE SMALLER OF THE TWO VECTORS IS LESS THAN	PLOT	55
C	FIVE TIMES THE TOTAL RANGE OF THE TWO VECTORS SCALED TOGETHER.	PLOT	56
C		PLOT	57
C	5) IF NP ≠ 2, ONLY LEFT-HAND TICK MARKS ARE GENERATED ON THE	PLOT	58



C	VERTICAL SIDE.	PLOT	59
C		PLOT	60
C	6) PAGE SKIP AND HEADING TITLES MUST BE GENERATED IN THE CALLING PROGRAM.	PLOT	61
C		PLOT	62
C		PLOT	63
C	7) IF ANY LOG OPTION IS CALLED, INITIAL AND INCREMENTAL VALUES EXPLICITLY SPECIFIED WITHOUT AUTOMATIC SCALING REQUEST (E.G., X0, DX, Y0, DY) ARE UNDERSTOOD TO BE ACTUAL LOG QUANTITIES.	PLOT	64
C		PLOT	65
C		PLOT	66
C		PLOT	67
C	8) IF MP DATA POINTS SAMPLED WITH A REPETITION MULTIPLE MULT EXCEEDS MM, MULT IS REDUCED TO ITS MAXIMUM ALLOWED VALUE.	PLOT	68
C		PLOT	69
C	INPUT PARAMETERS --	PLOT	70
C		PLOT	71
C	X0 = INITIAL VALUE FOR INDEPENDENT VARIABLE X.	PLOT	72
C		PLOT	73
C	DX = INCREMENTAL VALUE FOR THE INDEPENDENT VARIABLE X, CORRESPONDING TO A RESOLUTION OF THE X-AXIS INTO 50 INTERVALS, SPANNED BY A MAXIMUM OF 51 POINTS. TICK MARK INTERVAL = 5*MULT*DX	PLOT	74
C		PLOT	75
C		PLOT	76
C		PLOT	77
C		PLOT	78
C	Y0(I) = INITIAL VALUE FOR LEFT (I=1) AND RIGHT (I=2) VERTICAL SCALES.	PLOT	79
C		PLOT	80
C		PLOT	81
C	DY(I) = INCREMENTAL TICK MARK INTERVAL FOR THE LEFT (I=1) AND RIGHT HAND (I=2) VERTICAL SCALES.	PLOT	82
C		PLOT	83
C		PLOT	84
C		PLOT	85
C	X(I) = INDEPENDENT VARIABLE PROVIDED BY THE CALLING PROGRAM, USED ONLY WHEN DX = 0 AND AUTOMATIC X-AXIS SCALING IS REQUIRED. THE VECTOR X IS DESTROYED IN SOME SITUATIONS.	PLOT	86
C		PLOT	87
C		PLOT	88
C		PLOT	89
C		PLOT	90
C	MM = DIMENSION DECLARATOR FOR THE ARRAY Y IN THE CALLING PROGRAM: Y(MM,...).	PLOT	91
C		PLOT	92
C		PLOT	93
C	MP = NUMBER OF DATA POINTS TO BE SAMPLED FOR PLOT GENERATION.	PLOT	94
C		PLOT	95
C		PLOT	96
C	MULT = REPETITION FACTOR.	PLOT	97
C		PLOT	98
C	Y(I,J) = ARRAY, DIMENSIONED Y(MM,...) IN THE CALLING PROGRAM. PLOTS OF THE VECTORS J = 1,2,...,MP ARE GENERATED FROM DATA POINTS Y(I,J), Y(I+MULT,J), Y(I+2*MULT,J), ..., AND EACH POINT Y(I,J) IS ASSUMED TO CORRESPOND TO THE DEPENDENT VARIABLE X(I) = X0 + (I-1)*MULT*DX. IF DX = 0 AND AUTOMATIC X-AXIS SCALING IS REQUIRED, THE VECTOR X PROVIDED BY INPUT IS USED.	PLOT	99
C		PLOT	100
C		PLOT	101
C		PLOT	102
C		PLOT	103
C		PLOT	104
C		PLOT	105
C		PLOT	106
C		PLOT	107
C	CLEAR = LOGICAL VARIABLE	PLOT	108
C		PLOT	109
C	CLEAR = TRUE : NORMAL PLOT	PLOT	110
C	CLEAR = FALSE: HISTOGRAM	PLOT	111
C		PLOT	112
C	CENTER = LOGICAL VARIABLE TO CENTER PLOT ON PAGE.	PLOT	113
C		PLOT	114
C	SCALEX = LOGICAL VARIABLE TO SPECIFY AUTOMATIC SCALING OF	PLOT	115



C		HORIZONTAL AXIS.	PLOT	116
C	SCALEY	= LOGICAL VARIABLE TO SPECIFY AUTOMATIC SCALING OF VERTICAL AXIS.	PLOT	117
C			PLOT	118
C	SAME	= LOGICAL VARIABLE TO SPECIFY SAME SCALE ON THE LEFT AND RIGHT HAND VERTICAL AXES.	PLOT	119
C			PLOT	120
C	NAME(J)	= VECTOR OF WORDS (10 BCD CHARACTERS) TO LABEL THE JTH VECTOR PLOTTED.	PLOT	121
C			PLOT	122
C	NP	= NUMBER OF VECTORS TO BE PLOTTED (IF NP > 10, DEFAULT TO ONLY 10 PLOTS OCCURS.)	PLOT	123
C			PLOT	124
C	IP	= PLOT OPTION, FUNCTIONALLY SIMILAR TO THE USE OF THE FOUR ALTERNATE ENTRY POINTS:	PLOT	125
C			PLOT	126
C		PLOT (IP = 0) -- Y VS X.	PLOT	127
C		PYLOGX (IP = -1) -- LOGY VS X.	PLOT	128
C		PXLOGY (IP = 2) -- Y VS LOGX.	PLOT	129
C		PLOGLOG (IP = 1) -- LOGY VS LOGX.	PLOT	130
C			PLOT	131
C			PLOT	132
C			PLOT	133
C			PLOT	134
C			PLOT	135
C			PLOT	136
C			PLOT	137
C		.....	PLOT	138
C		DIMENSION Y(MULT,MH,1), YP(51,10), K(2,51), TICK(11), FORM(10),	PLOT	139
C		1 DY(1), X1(2), KX(2), DOT(10), XA(10), XB(10), CAPTION(10), Z0(3),	PLOT	140
C		2 DZ(3), X(MULT,1), Y0(1), NAME(1)	PLOT	141
C			PLOT	142
C		LOGICAL T1, T2, TEST, SCALEX, SCALEY, UNSCALE, CENTER, CLEAR,	PLOT	143
C		1 SAME, DIFF, SCALE, XLOG, YLOG	PLOT	144
C			PLOT	145
C		INTEGER DOT	PLOT	146
C		DATA DOT / 1H*, 1H., 1HX, 1H-, 1H., 1H0, 1HS, 1HE, 1H=, 1H, /	PLOT	147
C			PLOT	148
C		SET UP GENERAL CONTROL CONDITIONS --	PLOT	149
C			PLOT	150
C		ISW = 0	PLOT	151
C		TEST = (IP-1)*(IP+1)*(IP-2).EQ.0	PLOT	152
C		IF (TEST) ISW = IP	PLOT	153
C		GO TO 40	PLOT	154
C		ENTRY PYLOGX	PLOT	155
C		ISW = -1	PLOT	156
C		GO TO 40	PLOT	157
C		ENTRY PXLOGY	PLOT	158
C		ISW = 2	PLOT	159
C		GO TO 40	PLOT	160
C		ENTRY PLOGLOG	PLOT	161
C		ISW = 1	PLOT	162
C		40 NVEC = 10	PLOT	163
C		DXMULT = DX*MULT	PLOT	164
C		XLOG = IABS(ISW).EQ.1	PLOT	165
C		YLOG = ISW.GE.1	PLOT	166
C		NPLOT = NP	PLOT	167
C		IF (NPLOT.EQ.0) GO TO 99	PLOT	168
C		IF (NP.GT.NVEC) NPLOT = NVEC	PLOT	169
C		IF (.NOT.CLEAR) NPLOT = 1	PLOT	170
C		DIFF = .FALSE.	PLOT	171
C			PLOT	172

```

      IF (NPLOT.EQ.2) DIFF = .NOT.SAME
      MPLOT = MP
      SCALE = SCALEX.OR.(MPLOT.GT.51)
      IF(SCALE) MPLOT = 51
      MA = M1 = 1
      MB = M2 = MPLOT
      DO 50 I=1, NVEC
50  XA(I) = XB(I) = 0.0
      DZ(1) = DY(1)
      ZO(1) = YO(1)
      ZO(2) = DZ(2) = 0.0
      ZO(3) = X0
      DZ(3) = 5.*DXMULT
C
C      SET UP X-AXIS CONTROL --
C
      UNSCALE = .NOT.SCALE
      XMAX = XMIN = 0.
      IF (UNSCALE) GO TO 61
      IF(DX.EQ.0.0) GO TO 63
      X(1,1) = X0
      DO 54 I=2, MP
54  X(1,I) = X(1,I-1) * DXMULT
63  XMIN = X(1,1)
      XMAX = X(1,MP)
      EPS = 1.E-05*ABS(XMAX)
      XMAX = XMAX - EPS
      EPS = 1.E-05*ABS(XMIN)
      XMIN = XMIN + EPS
      IF(.NOT.XLOG) GO TO 61
      IF (XMIN.EQ.0.) XMIN = X(1,2)/10.
      IF (XMIN.LE.0.) GO TO 98
      XMIN = ALOG10(XMIN)
      XMAX = ALOG10(XMAX)
61  CALL AXIS (SCALE, XMAX, XMIN, ZO(3), DZ(3), XI)
C
C      SET UP PLOTTING ARRAYS YP --
C
      XI = ZO(3)
      DXI = DZ(3)/5.0
      DO 37 M=1, MPLOT
      XP = XI
      IF(XLOG) XP = 10.0**XI
      IF(UNSCALE) GO TO 53
      IF (XP.GT.X(1,1)) GO TO 55
      M1 = M
      MA = M1+1
55  IF (XP.LT.X(1,MP)) MB = M
      M2 = MB+1
53  DO 33 I=1, NPLOT
      YP(M,I) = Y(1,M,I)
      IF(UNSCALE) GO TO 33
      CALL INTERP (2, XP, YP(M,I), X, Y(1,1,I), MULT, MP)
33  CONTINUE
37  XI = XI + DXI
      IF (M2.GT.MPLOT) M2 = MPLOT
      IF(.NOT.YLOG) GO TO 45

```

PLOT	173
PLOT	174
PLOT	175
PLOT	176
PLOT	177
PLOT	178
PLOT	179
PLOT	180
PLOT	181
PLOT	182
PLOT	183
PLOT	184
PLOT	185
PLOT	186
PLOT	187
PLOT	188
PLOT	189
PLOT	190
PLOT	191
PLOT	192
PLOT	193
PLOT	194
PLOT	195
PLOT	196
PLOT	197
PLOT	198
PLOT	199
PLOT	200
PLOT	201
PLOT	202
PLOT	203
PLOT	204
PLOT	205
PLOT	206
PLOT	207
PLOT	208
PLOT	209
PLOT	210
PLOT	211
PLOT	212
PLOT	213
PLOT	214
PLOT	215
PLOT	216
PLOT	217
PLOT	218
PLOT	219
PLOT	220
PLOT	221
PLOT	222
PLOT	223
PLOT	224
PLOT	225
PLOT	226
PLOT	227
PLOT	228
PLOT	229



C	M1 = MA	PLOT	230
	M2 = MB	PLOT	231
	DO 34 I=1, NPLOT	PLOT	232
	DO 34 M=M1, M2	PLOT	233
	IF (YP(M,I).LE.0.) GO TO 98	PLOT	234
	34 YP(M,I) = ALOG10(YP(M,I))	PLOT	235
C		PLOT	236
C	ESTABLISH MAXIMUM AND MINIMUM VALUES FOR EACH VECTOR --	PLOT	237
C		PLOT	238
	45 DO 1 I=1, NPLOT	PLOT	239
	XA(I) = XB(I) = YP(MA,I)	PLOT	240
	DO 2 M = MA, MB	PLOT	241
	YMI = YP(M,I)	PLOT	242
	IF (XA(I).GT.YMI) XA(I) = YMI	PLOT	243
	2 IF (XB(I).LT.YMI) XB(I) = YMI	PLOT	244
	IF (XA(I).GE.0.) GO TO 1	PLOT	245
	IF (XB(I).LE.0.) GO TO 1	PLOT	246
	A = ABS(XA(I) + XB(I))	PLOT	247
	B = ABS(XA(I) - XB(I))	PLOT	248
	IF ((B-A).GT.(A+B)/100.) GO TO 1	PLOT	249
	B = XB(I)	PLOT	250
	IF (B.GT.A) XA(I) = 0.	PLOT	251
	IF (B.LT.A) XB(I) = 0.	PLOT	252
	1 CONTINUE	PLOT	253
	NSCALE = 1	PLOT	254
C		PLOT	255
C	FIND LARGEST XB(I) AND SMALLEST XA(I) --	PLOT	256
C		PLOT	257
	IF (NPLOT.EQ.1) GO TO 18	PLOT	258
	NSCALE = 2	PLOT	259
	XMIN = XA(1)	PLOT	260
	XMAX = XB(1)	PLOT	261
	DO 23 I = 2, NPLOT	PLOT	262
	IF (XA(I).LT.XMIN) XMIN = XA(I)	PLOT	263
	23 IF (XB(I).GT.XMAX) XMAX = XB(I)	PLOT	264
	EPS = 1.0E-05*ABS(XMAX-XMIN)	PLOT	265
	XMIN = XMIN + EPS	PLOT	266
	XMAX = XMAX - EPS	PLOT	267
C		PLOT	268
C	SET UP Y-AXIS CONTROL (FOR SINGLE OR DOUBLE SCALING) --	PLOT	269
C		PLOT	270
	IF (.NOT.DIFF) GO TO 18	PLOT	271
	Z0(2) = Y0(2)	PLOT	272
	DZ(2) = DY(2)	PLOT	273
	10 A = XB(1) - XA(1)	PLOT	274
	B = XB(2) - XA(2)	PLOT	275
	A = AMAX1(XB(1),A)	PLOT	276
	B = AMAX1(XB(2),B)	PLOT	277
	DS = AMIN1(A,B)	PLOT	278
	XMULT = 10.	PLOT	279
	IF (NPLOT.EQ.2) DIFF = DIFF.OR.((XMAX-XMIN).GT.XMULT*DS)	PLOT	280
	IF (DIFF) GO TO 18	PLOT	281
	NSCALE = 1	PLOT	282
	DZ(2) = DZ(1)	PLOT	283
	Z0(2) = Z0(1)	PLOT	284
	XA(1) = XMIN	PLOT	285
		PLOT	286



	XB(1) = XMAX	PLOT	287
	XA(2) = XB(2) = 0.0	PLOT	288
C		PLOT	289
18	DO 15 I = 1, NSCALE	PLOT	290
	SCALE = SCALEY.OR.(DZ(I).EQ.0.0)	PLOT	291
	IF (.NOT.CLEAR) XA(I) = 0.	PLOT	292
	CALL AXIS (SCALE, XB(I), XA(I), Z0(I), DZ(I), XI)	PLOT	293
	M0 = 0	PLOT	294
	IF (.NOT.SCALE) GO TO 29	PLOT	295
	XMAX = Z0(I) + 10.*DZ(I)	PLOT	296
	M0 = 0.5*((XMAX-XB(I))-(XA(I)-Z0(I)))/DZ(I)	PLOT	297
	IF ((XI.EQ.0.).AND.(Z0(I).EQ.0.)) M0 = 0	PLOT	298
29	DS = M0*DZ(I)	PLOT	299
	IF (XI.EQ.0.) GO TO 31	PLOT	300
	XI = XI - DS	PLOT	301
	Z0(I) = Z0(I) + DS	PLOT	302
31	Z0(I) = Z0(I) - DS	PLOT	303
15	XA(I) = XI	PLOT	304
	IF (.NOT.DIFF) XA(2) = XA(1)	PLOT	305
C	XA(1), XA(2) ARE DC BASELINE VALUES AT THIS POINT.	PLOT	306
C		PLOT	307
	DO 38 I=1, NPLLOT	PLOT	308
	XI = XA(1)	PLOT	309
	IF (I.EQ.2) XI = XA(2)	PLOT	310
	DO 38 M=M1,M2	PLOT	311
38	YP(M,I) = YP(M,I) - XI	PLOT	312
C		PLOT	313
	DO 11 I=1, NSCALE	PLOT	314
11	XI(I) = Z0(I) + 11.0*DZ(I)	PLOT	315
	KX(1) = KX(2) = -1	PLOT	316
	EPS = 0.02	PLOT	317
	EFA = 0.5 + EPS	PLOT	318
C		PLOT	319
C	DETERMINE LOCATION OF HORIZONTAL AXES --	PLOT	320
C		PLOT	321
	DO 9 I = 1, NSCALE	PLOT	322
	IF (DZ(I).EQ.0.0) GO TO 9	PLOT	323
	KX(I) = -5.0*Z0(I)/DZ(I)*1.00001	PLOT	324
9	CONTINUE	PLOT	325
C		PLOT	326
C	CONVERT PLOTTING ARRAY YP TO NORMALIZE VALUES BETWEEN 0 AND 50	PLOT	327
C	(CORRESPONDING TO LINES 1,2,3,...,51 FOR EACH PLOTTED PAGE) --	PLOT	328
C		PLOT	329
	DO 4 I = 1, NPLLOT	PLOT	330
	DXI = DZ(I)	PLOT	331
	X0I = Z0(I)	PLOT	332
	IF (I.NE.2) GO TO 88	PLOT	333
	IF (.NOT.DIFF) GO TO 88	PLOT	334
	X0I = Z0(2)	PLOT	335
	DXI = DZ(2)	PLOT	336
88	CONTINUE	PLOT	337
	DO 4 M = M1,M2	PLOT	338
4	YP(M,I) = ((YP(M,I)-X0I)/DXI)*5.0	PLOT	339
C		PLOT	340
C	DETERMINE NUMBER OF HORIZONTAL ELEMENTS PER LINE OF PLOT --	PLOT	341
C		PLOT	342
	IMAX = (M2 + 3)/5	PLOT	343

IMAX1 = IMAX + 1	PLOT	344
MAX = 5 * IMAX + 1	PLOT	345
MAX2 = 2 * MAX	PLOT	346
N = 68 - MAX	PLOT	347
IF(.NOT.DIFF) N = N * 5	PLOT	348
IF(.NOT.CENTER) N = 9	PLOT	349
N2 = N - 2	PLOT	350
ENCODE(70,100,FORM) N, N2, MAX2, N	PLOT	351
100 FORMAT (1H(12,7HX102A1),7H(1H+1PG12,3H,4+13,10HX,1PG14,4)5X,4H(1H+	PLOT	352
1 12, 20HX,7X,A1,5H --- ,A10))	PLOT	353
ENCODE(90,101,CAPTION) N, N, N	PLOT	354
101 FORMAT(4H(1H+,12,24HX4X17H * --- LHS AXIS),5H(1H+,12, 23HX,22X,	PLOT	355
* * (*,E12,4,*)*, 4H(1H+,12,24HX4X17H * --- RHS AXIS))	PLOT	356
C PLOT 51 LINES --	PLOT	357
C	PLOT	358
C	PLOT	359
LINE1 = 5	PLOT	360
DO 6 L=1, 51	PLOT	361
LL = L-1	PLOT	362
XLINE = LINE = 51-L	PLOT	363
T1 = (LINE.EQ.KX(1)).OR.(LINE.EQ.KX(2))	PLOT	364
T2 = (LL/50)*50.EQ.LL	PLOT	365
TEST = T1.OR.T2	PLOT	366
DO 5 M=1, MAX	PLOT	367
K(2,M) = 1H	PLOT	368
DO 42 I=1, NPLOT	PLOT	369
IF(K(1,M).EQ.DOT(I)) GO TO 44	PLOT	370
42 CONTINUE	PLOT	371
K(1,M) = 1H	PLOT	372
44 IF(CLEAR) K(1,M) = 1H	PLOT	373
5 IF(TEST) K(1,M) = K(2,M) = 1H-	PLOT	374
IF(.NOT.TEST) GO TO 14	PLOT	375
DO 8 I=1, 11	PLOT	376
I1 = 1 + 5*(I-1)	PLOT	377
8 K(1,I1) = 1HI	PLOT	378
14 K(1,1) = K(1,MAX) = 1HI	PLOT	379
IF(L.EQ.51) GO TO 22	PLOT	380
T2 = 5*(LL/51).EQ.LL	PLOT	381
IF(.NOT.(T1.OR.T2)) GO TO 24	PLOT	382
K(2,1) = K(2,MAX-1) = 1H-	PLOT	383
DO 62 I=1, NPLOT	PLOT	384
IF(K(1,M).EQ.DOT(I)) GO TO 64	PLOT	385
62 CONTINUE	PLOT	386
K(1,2) = 1H-	PLOT	387
64 DO 66 I=1, NPLOT	PLOT	388
IF(K(1,MAX-1).EQ.DOT(I)) GO TO 24	PLOT	389
66 CONTINUE	PLOT	390
K(1,MAX-1) = 1H-	PLOT	391
24 DO 3 I=1, NPLOT	PLOT	392
DO 3 M=M1, M2	PLOT	393
IF(ABS(XLINE-YP(M,I)).LT.ETA) K(1,M) = DOT(I)	PLOT	394
IF((M.EQ.M2).OR.(.NOT.CLEAR)) GO TO 3	PLOT	395
XAVG = (YP(M,I)+YP(M+1,I))/2.0	PLOT	396
IF(ABS(XLINE-XAVG).LT.ETA) K(2,M) = DOT(I)	PLOT	397
X1AVG = 0.5*(XAVG + YP(M,I))	PLOT	398
X2AVG = 0.5*(XAVG + YP(M+1,I))	PLOT	399
IF(ABS(XLINE-X1AVG).LT.ETA) K(2,M) = DOT(I)	PLOT	400



```

IF(ABS(XLINE-X2AVG).LT.ETA) K(2,M) = DOT(I)
XMAX = AMAX1(X1AVG,X2AVG)
XMIN = AMIN1(X1AVG,X2AVG)
IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(2,M) = DOT(I)
XMAX = AMAX1(X1AVG,YP(M,I))
XMIN = AMIN1(X1AVG,YP(M,I))
IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(1,M) = DOT(I)
XMAX = AMAX1(X2AVG,YP(M+1,I))
XMIN = AMIN1(X2AVG,YP(M+1,I))
IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(1,M+1) = DOT(I)
3 CONTINUE
22 K(2,MAX) = 1H
M0 = 12
TEST = (L.LT.49).AND.(L.GT.44)
IF(.NOT.TEST) GO TO 20
IF(.NOT.DIFF.AND.XA(1).EQ.0.0) GO TO 20
IF(XA(1).NE.0.0).OR.(XA(2).NE.0.0) M0 = 21
GO TO 21
20 IF (NPLOT.EQ.1) GO TO 13
IF (LINE1.NE.L) GO TO 13
L1 = L-4
IF (L1.GT.NPLOT) GO TO 13
LINE1 = LINE1 + 1
21 DO 19 M=3, M0
19 K(1,M) = K(2,M) = 1H
13 WRITE(6,FORM(1)) (K(1,M), K(2,M), M=1, MAX)
IF (L.LE.4) GO TO 16
IF (LINE1.EQ.(L+1)) WRITE (6,FORM(5)) DOT(L1), NAME(L1)
16 IF(.NOT.DIFF.AND.XA(1).EQ.0.0) GO TO 25
IF(L.NE.46) GO TO 28
WRITE(6,CAPTION(1))
IF(XA(1).NE.0.0) WRITE(6,CAPTION(4)) XA(1)
28 IF(L.NE.47) GO TO 25
IF(.NOT.DIFF) GO TO 25
WRITE(6,CAPTION(7))
IF(XA(2).NE.0.0) WRITE(6,CAPTION(4)) XA(2)
25 IF((LL/5)*5.NE.LL) GO TO 6
DO 17 I = 1,NSCALE
X1(I) = X1(I) - DZ(I)
IF(ABS(X1(I)).LT.0.001*DZ(I)) X1(I) = 0.0
17 XB(I) = X1(I)
WRITE(6,FORM(2)) (XB(I), I=1, NSCALE)
6 CONTINUE
N8 = N-8
IF(UNSCALE.AND.(DX.EQ.0.0)) GO TO 30
AXMAX = ABS(Z0(3) + IMAX*DZ(3))
IF(AXMAX.EQ.0.0) AXMAX = 10.0*ABS(DZ(3))
MX = A = ALOG10(AXMAX)
MX = MX+1
IF (A.LT.0.) MX = MX-1
SX = 10.0**MX
IF(MX.LT.4).AND.(MX.GE.0) SX = 1.0
DO 7 I=1, IMAX1
TICK(I) = Z0(3) + (I-1)*DZ(3)
7 TICK(I) = TICK(I)/SX
SX = TICK(2)
IF (SX.EQ.0.) SX = TICK(3)

```

```

PLOT 401
PLOT 402
PLOT 403
PLOT 404
PLOT 405
PLOT 406
PLOT 407
PLOT 408
PLOT 409
PLOT 410
PLOT 411
PLOT 412
PLOT 413
PLOT 414
PLOT 415
PLOT 416
PLOT 417
PLOT 418
PLOT 419
PLOT 420
PLOT 421
PLOT 422
PLOT 423
PLOT 424
PLOT 425
PLOT 426
PLOT 427
PLOT 428
PLOT 429
PLOT 430
PLOT 431
PLOT 432
PLOT 433
PLOT 434
PLOT 435
PLOT 436
PLOT 437
PLOT 438
PLOT 439
PLOT 440
PLOT 441
PLOT 442
PLOT 443
PLOT 444
PLOT 445
PLOT 446
PLOT 447
PLOT 448
PLOT 449
PLOT 450
PLOT 451
PLOT 452
PLOT 453
PLOT 454
PLOT 455
PLOT 456
PLOT 457

```



SX = ABS(SX)	PLOT	458
M0 = ALOG10(SX)	PLOT	459
IF(M0.GE.0) N8 = N8 + 1	PLOT	460
ENCODE(30,102,FORM) N8	PLOT	461
102 FORMAT(2H(/,I2,2HX,,8H11F10.2))	PLOT	462
WRITE(6,FORM) (TICK(I), I=1,IMAX1)	PLOT	463
IF((MX.GE.4).OR.(MX.LT.0)) WRITE(6,103) MX	PLOT	464
103 FORMAT(/68X,7H(X 10**,I3,*)*)	PLOT	465
GO TO 99	PLOT	466
30 ENCODE(20,300,FORM) N	PLOT	467
300 FORMAT(2H(/,I2,11HX,*0*10I10))	PLOT	468
WRITE(6,FORM) (M, M=5, MAX, 5)	PLOT	469
GO TO 99	PLOT	470
98 WRITE(6,104)	PLOT	471
104 FORMAT(10(/,40X,*AN ATTEMPT WAS MADE TO PLOT VALUES ≤ 0 WITH LOG	PLOT	472
1 OPTION*10(/))	PLOT	473
99 RETURN	PLOT	474
END	PLOT	475
SUBROUTINE AXIS (SCALE, XMAX, XMIN, X0, DX, XDC)	AXIS	2
LOGICAL SCALE	AXIS	3
C	AXIS	4
C	AXIS	5
C	AXIS	6
C	AXIS	7
C	AXIS	8
C	AXIS	9
C	AXIS	10
C	AXIS	11
C	AXIS	12
C	AXIS	13
C	AXIS	14
C	AXIS	15
C	AXIS	16
C	AXIS	17
C	AXIS	18
C	AXIS	19
C	AXIS	20
C	AXIS	21
C	AXIS	22
C	AXIS	23
C	AXIS	24
C	AXIS	25
C	AXIS	26
C	AXIS	27
C	AXIS	28
C	AXIS	29
C	AXIS	30
C	AXIS	31
C	AXIS	32
C	AXIS	33
C	AXIS	34
C	AXIS	35
C	AXIS	36
C	AXIS	37
C	AXIS	38
C	AXIS	39
C	AXIS	40
C	AXIS	41
C	AXIS	42
C	AXIS	43
C	AXIS	44
C	AXIS	45
C	AXIS	46
C	AXIS	47
C	AXIS	48
C	AXIS	49
C	AXIS	50
C	AXIS	51
C	AXIS	52
C	AXIS	53
C	AXIS	54
C	AXIS	55
C	AXIS	56

```

SUBROUTINE AXIS (SCALE, XMAX, XMIN, X0, DX, XDC)
LOGICAL SCALE
.....
XMAX AND XMIN ARE THE LARGEST AND SMALLEST VALUES OF THE VECTOR TO
BE SCALED WITH CONVENIENT ORIGIN X0 AND TICK MARK SPACING DX. X1
IS SET EQUAL TO ZERO UNLESS THE AC RANGE 10*DX ≤ X0/1000. IN THAT
CASE XDC IS SET EQUAL TO X0. X0 IS SET EQUAL TO ZERO, AND XMAX AND
XMIN ARE REDUCED BY AN AMOUNT XDC. THUS, XDC CORRESPONDS TO A DC
BASELINE THAT IS RETURNED NONZERO ONLY IF THE RANGE OF THE PLOT IS
VERY SMALL RELATIVE TO THE ABSOLUTE MAGNITUDE OF PLOTTED VALUES.
.....
IF (.NOT.SCALE) GO TO 3
X0 IS CONVENIENT ORIGIN, AND RANGE 10*DX IS 1, 2, OR 5 TIMES SOME
POWER OF 10 --
IF (ABS(XMAX-XMIN).GT.1.0E-10*(XMAX*XMIN)) GO TO 1
XMIN = 0.
IF (XMAX.EQ.0.) XMAX = 1.0
1 XA = XMIN
4 B = XMAX - XA
M = A = ALOG10(B)
IF (A.LT.0.) M = M-1
DS = 10.**M
M = B/DS + 1.
B = 2.
IF (M.GT.2) B = 5.
IF (M.GT.5) B = 10.
B = B*DS
DS = B/10.
M = A = XA/DS
IF (A.LT.0.) M = M-1
XA = M*DS
IF ((XA*B).LT.XMAX) GO TO 4
DX = DS
X0 = XA
C CALCULATE DC BASELINE VALUE XDC --
3 XDC = 0.
IF (ABS(X0).LT.1.E 04*DX) GO TO 2
XMIN = XMIN - X0
XMAX = XMAX - X0
XDC = X0
X0 = 0.
2 IF (.NOT.SCALE) GO TO 99
ALLOW DATA TO BE PLOTTED WITH ZERO ORIGIN (AT TOP OR BOTTOM) IF
POSSIBLE --
X1 = X0 + 10.*DX
IF ((XMAX.LT.10.*DX).AND.(X0.GT.0.)) X0 = 0.
IF ((XMIN.GT.-10.*DX).AND.(X1.LT.0.)) X0 = -10.*DX
99 RETURN
END

```

	SUBROUTINE INTERP (IDEG, XP, YP, X, Y, MULT, N)	INTERP	2
C	.....	INTERP	3
C		INTERP	4
C		INTERP	5
C	THIS SUBROUTINE WILL INTERPOLATE A VECTOR Y, DEFINED AS A FUNCTION	INTERP	6
C	OF THE VECTOR X, TO PRODUCE THE VALUE YP THAT CORRESPONDS TO XP.	INTERP	7
C	IDEG = 1, 2, ... SPECIFIES LINEAR, QUADRATIC, ... ETC. INTERPOLA-	INTERP	8
C	TION. THE VECTORS X AND Y PROVIDED BY THE CALLING PROGRAM ARE	INTERP	9
C	SAMPLED WITH A REPETITION INDEX = MULT. AND N POINTS DEFINED BY	INTERP	10
C	I = 1, (1+MULT), ..., (1 + (N-1)*MULT) WILL BE UTILIZED. THUS,	INTERP	11
C	THE VECTORS X, Y MUST BE DIMENSIONED AT LEAST (1+(N-1)*MULT)	INTERP	12
C	IN THE CALLING PROGRAM. VALUES FOR THE INDEPENDENT VECTOR X MUST	INTERP	13
C	BE IN ASCENDING ORDER. XP NEED NOT LIE WITHIN THE RANGE X(1),...	INTERP	14
C	X(1+(N-1)*MULT), BUT IF IT FALLS OUTSIDE, LINEAR INTERPOLATION IS	INTERP	15
C	GIVEN. (THE DEFAULT TO LINEAR INTERPOLATION CAN BE REMOVED BY	INTERP	16
C	DELETION OF ONE CARD BELOW.) NORMAL USAGE IS MULT = 1.	INTERP	17
C	.....	INTERP	18
C		INTERP	19
C	DIMENSION T(20), X(MULT,1), Y(MULT,1)	INTERP	20
	INT = IDEG	INTERP	21
	IF (INT.LE.0) INT = 1	INTERP	22
	IF (INT.GE.N) INT = N-1	INTERP	23
	IF (N.LT.2) GO TO 10	INTERP	24
	DO 4 I = 1,N	INTERP	25
	J = I	INTERP	26
	IF (XP.LE.X(1,I)) GO TO 1	INTERP	27
	4 CONTINUE	INTERP	28
	1 CONTINUE	INTERP	29
C	DEFAULT TO LINEAR INTERPOLATION IF XP LIES OUTSIDE RANGE OF X --	INTERP	30
	IF ((J.EQ.1).OR.(J.EQ.N)) INT = 1	INTERP	31
	K = INT	INTERP	32
	INT = INT + 1	INTERP	33
	J = J - INT/2	INTERP	34
	J = MAX0(J,1)	INTERP	35
	J = MIN0(J,N-K)	INTERP	36
	JK = J.K	INTERP	37
	DO 2 I = J,JK	INTERP	38
	KK = I-J+1	INTERP	39
	T(KK) = Y(1,I)	INTERP	40
	2 T(KK+INT) = X(1,I) - XP	INTERP	41
	DO 3 I = 1,K	INTERP	42
	IP1 = I + 1	INTERP	43
	DO 3 JJ = IP1,INT	INTERP	44
	3 T(JJ) = (T(I)*T(JJ+INT)-T(JJ)*T(I+INT))/(X(1,JJ+J-1)-X(1,I+J-1))	INTERP	45
	YP = T(INT)	INTERP	46
	RETURN	INTERP	47
	10 YP = Y(1,1)	INTERP	48
	RETURN	INTERP	49
	END	INTERP	50
		INTERP	51
	SUBROUTINE SIMPSON (F, M, H, ANS)	SIMPSON	2
C	.....	SIMPSON	3
C		SIMPSON	4
C		SIMPSON	5
C	THIS SUBROUTINE INTEGRATES A FUNCTION F DEFINED AS A VECTOR OVER N =	SIMPSON	6
C	2M INTERVALS, F(1), F(2), F(3), ..., F(2M+1). THE WIDTH OF THE SUB-	SIMPSON	7
C	INTERVALS IS H, AND THE ANSWER IS RETURNED IN ANS.	SIMPSON	8
C	.....	SIMPSON	9
C		SIMPSON	10
C		SIMPSON	11
C	DIMENSION F(1)	SIMPSON	12
	N = M * M	SIMPSON	13
	SUM = F(N+1) - F(1)	SIMPSON	14
	DO 1 I = 1,N+2	SIMPSON	15
	II = I-1	SIMPSON	16
	Y = F(I) * F(II) * F(II)	SIMPSON	17
	1 SUM = SUM + Y * Y	SIMPSON	18
	ANS = H*SUM/3.	SIMPSON	19
	RETURN	SIMPSON	20
	END	SIMPSON	21



	SUBROUTINE EDITOR (INPUT, LIST)	EDITOR	2
C		EDITOR	3
C	.....	EDITOR	4
C		EDITOR	5
C	THIS SUBROUTINE READS AN INPUT CARD FILE TO THE EOF, AND WRITES IT	EDITOR	6
C	ONTO TAPE 5. IF LIST = .TRUE. IS SPECIFIED, IT ALSO PRODUCES A	EDITOR	7
C	WRITTEN OUTPUT LISTING OF THE INPUT CARD IMAGES.	EDITOR	8
C		EDITOR	9
C	.....	EDITOR	10
C		EDITOR	11
	DIMENSION IMAGE(8)	EDITOR	12
	LOGICAL LIST	EDITOR	13
	DATA SKIP, K / 5H(1H1), 1234567890 /	EDITOR	14
	REWIND 5	EDITOR	15
	KARD = 0	EDITOR	16
	CALL DATE (MONTH)	EDITOR	17
	1 READ (INPUT,150) IMAGE	EDITOR	18
	IF (EOF(INPUT)) 5,3	EDITOR	19
	3 IF (MOD(KARD,40).NE.0) GO TO 2	EDITOR	20
	IF (.NOT.LIST) GO TO 2	EDITOR	21
	IF (KARD.EQ.0) GO TO 4	EDITOR	22
	WRITE (6,110) (K, I = 1,8), (I, I = 1,8)	EDITOR	23
	WRITE (6,120)	EDITOR	24
	120 FORMAT (/21X,*CONTINUED*)	EDITOR	25
	4 WRITE (6,SKIP)	EDITOR	26
	WRITE (6,300) MONTH	EDITOR	27
	300 FORMAT (/47X,*SUMMARY OF CARD IMAGES FOR INPUT DATA DECK*/60X,*ID	EDITOR	28
	DATE:*.A9,*)*//)	EDITOR	29
	WRITE (6,100) (I, I = 1,8), (K, I = 1,8)	EDITOR	30
	100 FORMAT (22X,*CARD*,13X,8(11,9X)/23X,*NO.*,4X,8110//21X,92(1H-)/)	EDITOR	31
	2 KARD = KARD + 1	EDITOR	32
	WRITE (5,150) IMAGE	EDITOR	33
	150 FORMAT (8A10)	EDITOR	34
	IF (LIST) WRITE (6,200) KARD, IMAGE	EDITOR	35
	200 FORMAT (22X,13,* ....*,8A10)	EDITOR	36
	GO TO 1	EDITOR	37
C		EDITOR	38
	5 ENDFILE 5	EDITOR	39
	REWIND 5	EDITOR	40
	IF (KARD.EQ.0) GO TO 99	EDITOR	41
	IF (.NOT.LIST) GO TO 99	EDITOR	42
	WRITE (6,110) (K, I = 1,8), (I, I = 1,8)	EDITOR	43
	110 FORMAT (/21X,92(1H-)//30X,8110/39X,8(11,9X))	EDITOR	44
	99 RETURN	EDITOR	45
	END	EDITOR	46



	SUBROUTINE COVER (TITLE, NPAGE)	COVER	2
	DIMENSION TITLE(3), MESSAGE(10,3)	COVER	3
	INTEGER BLANK, TITLE	COVER	4
C	.....	COVER	5
C	.....	COVER	6
C	THIS SUBROUTINE WILL GENERATE NPAGE SEQUENTIAL TITLE PAGES FOR THE	COVER	7
C	OUTPUT PRINTOUT.	COVER	8
C	.....	COVER	9
C	.....	COVER	10
C	.....	COVER	11
	BLANK = 1H	COVER	12
	N = 0	COVER	13
	DO 2 I = 1,3	COVER	14
	IF (TITLE(I).EQ.BLANK) GO TO 2	COVER	15
	N = N+1	COVER	16
	TITLE(N) = TITLE(I)	COVER	17
	DECODE (10,100,TITLE(N)) (MESSAGE(J,N), J = 1,10)	COVER	18
100	FORMAT (10A1)	COVER	19
	2 CONTINUE	COVER	20
C	NSKIP = 3 + 6*(3-N)	COVER	21
	ENCODE (10,150,KONTROL) NSKIP	COVER	22
150	FORMAT (1H1,12)	COVER	23
	NSKIP = KONTROL	COVER	24
C	DO 1 K = 1,NPAGE	COVER	25
	J = 0	COVER	26
	KONTROL = NSKIP	COVER	27
3	IF (J.EQ.N) GO TO 1	COVER	28
	J = J+1	COVER	29
	CALL HEADINX (1HX, KONTROL, MESSAGE(1,J))	COVER	30
	KONTROL = BLANK	COVER	31
	GO TO 3	COVER	32
1	WRITE (6,200)	COVER	33
C	200 FORMAT (////	COVER	34
	1/39X,52H	COVER	35
	2/39X,52H	COVER	36
	3/39X,52H	COVER	37
	4/39X,52H	COVER	38
	5/39X,52H	COVER	39
	6/39X,52H	COVER	40
	7/39X,52H	COVER	41
	8/39X,52H	COVER	42
	9/39X,52H	COVER	43
	\$/39X,52H	COVER	44
	1/39X,52H	COVER	45
	2/39X,52H	COVER	46
	3/39X,52H	COVER	47
	.....)	COVER	48
C	RETURN	COVER	49
	END	COVER	50
		COVER	51
		COVER	52
		COVER	53
		COVER	54

	SUBROUTINE HEADINX (JSYMB, JPAGE, MESSAGE)	HEADINX	2
C		HEADINX	3
C	.....	HEADINX	4
C		HEADINX	5
C	THE CALL SEQUENCE FOR THIS SUBROUTINE IS EQUIVALENT TO THAT SUPPLIED	HEADINX	6
C	BY THE CDC CYBERNET SYSTEM (WITH THE SAME NAME) FOR THE GENERATION OF	HEADINX	7
C	BLOCK HEADINGS. HOWEVER, THE PRESENT VERSION HAS TWO ADDITIONAL AD-	HEADINX	8
C	VANTAGES:	HEADINX	9
C		HEADINX	10
C	1) THE 10-BCD CHARACTER WORD STORED IN THE VECTOR MESSAGE(I)	HEADINX	11
C	(I = 1,2,...,10) IS AUTOMATICALLY CENTERED ON THE PRINTED	HEADINX	12
C	LINE, AND	HEADINX	13
C		HEADINX	14
C	2) THE CARRIAGE CONTROL SYMBOL *JPAGE* CAN ACCEPT MORE GENERAL	HEADINX	15
C	SPECIFICATIONS TO CONTROL THE SPACING OF THE PRINTED LINE.	HEADINX	16
C	JPAGE CAN HAVE THE USUAL FORMAT IHS (WHERE S = 0, 1, 2, +,	HEADINX	17
C	ETC), OR IT CAN HAVE A MORE GENERAL FORMAT JHSNN, WHERE NN	HEADINX	18
C	IS A TWO-DIGIT NUMBER. AFTER THE PAGE CONTROL S IS EXECUTED,	HEADINX	19
C	TWO LINES ARE AUTOMATICALLY SPACED, FOLLOWED BY AN ADDITIONAL	HEADINX	20
C	NN LINES. THE REMAINING SEVEN BITS OF *JPAGE* ARE IGNORED.	HEADINX	21
C		HEADINX	22
C	UNLIKE THE CDC SUBROUTINE, THE PLOTTING SYMBOL *JSYMB* IS IGNORED,	HEADINX	23
C	AND THE CHARACTER X IS ALWAYS USED.	HEADINX	24
C		HEADINX	25
C	.....	HEADINX	26
C		HEADINX	27
C	DIMENSION KAR(10,50), FORM(3), MESSAGE(10), LETTER(50), NUM(10),	HEADINX	28
C	1 IMAGE(10), KK(500)	HEADINX	29
C		HEADINX	30
C	INTEGER BLANK	HEADINX	31
C	EQUIVALENCE (KAR, KK)	HEADINX	32
C		HEADINX	33
C	DATA LETTER / IMA, IMB, IMC, IMD, IME, IMF, IMG, IMH, IMI, IMJ,	HEADINX	34
C	1 IMK, IML, IMM, IMN, IMO, IMP, IMQ, IMR, IMS, IMT, IMU, IMV, IMW,	HEADINX	35
C	2 IMX, IMY, IMZ, IH1, IH2, IH3, IH4, IH5, IH6, IH7, IH8, IH9, IHO,	HEADINX	36
C	3 IH., IH-, IH/, IH., IH:, IH\$, IH=, IH(, IH), IH<, IH>, IH!, IH!.	HEADINX	37
C	4 IH /	HEADINX	38
C		HEADINX	39
C	DATA (KK(I), I = 1,150)	HEADINX	40
C	5 / 10H XXXXXXXX, 10HXXXXXXXXXXXX, 3*10HXX XX,	HEADINX	41
C	1 2*10HXXXXXXXXXXXX, 3*10HXX XX, 10HXXXXXXXXXXXX, 10HXXXXXXXXXXXX,	HEADINX	42
C	2 2*10HXX XX, 2*10HXXXXXXXXXXXX, 2*10HXX XX, 10HXXXXXXXXXXXX,	HEADINX	43
C	3 10HXXXXXXXXXXXX, 10H XXXXXXXX, 10HXXXXXXXXXXXX, 10HXX XX,	HEADINX	44
C	4 4*2HXX, 10HXX XX, 10HXXXXXXXXXXXX, 10H XXXXXXXX, 9HXXXXXXXXXXXX,	HEADINX	45
C	5 10HXXXXXXXXXXXX, 6*10HXX XX, 10HXXXXXXXXXXXX, 9HXXXXXXXXXXXX,	HEADINX	46
C	6 2*10HXXXXXXXXXXXX, 2*2HXX, 2*6HXXXXXXXX, 2*2HXX, 4*10HXXXXXXXXXXXX,	HEADINX	47
C	7 2*2HXX, 2*6HXXXXXXXX, 4*2HXX, 9H XXXXXXXX, 10HXXXXXXXXXXXX, 10HXX XX,	HEADINX	48
C	8 2HXX, 2*10HXX XXXXX, 2*10HXX XX, 10HXXXXXXXXXXXX, 9H XXXXXXXX,	HEADINX	49
C	9 4*10HXX XX, 2*10HXXXXXXXXXXXX, 4*10HXX XX, 2*9H XXXXXXXX,	HEADINX	50
C	5 6*6H XX, 2*9H XXXXXXXX, 2*10H XXXXXXXX, 4*8H XX, 2*8HXX	HEADINX	51
C	1XX, 8HXXXXXXXXXXXX, 7H XXXXXXXX, 10HXX XX, 9HXX XX, 7HXX XX,	HEADINX	52
C	2 5HXX XX, 5HXXXXX, 6HXX XX, 7HXX XX, 8HXX XX, 9HXX XX,	HEADINX	53
C	3 10HXX XX, 8*2HXX, 2*10HXXXXXXXXXXXX, 10HXX XX,	HEADINX	54
C	4 10HXX XXX, 10HXXXX XXXX, 2*10HXX XXXX XX, 2*10HXX XX XX,	HEADINX	55
C	5 4*10HXX XX, 10HXX XX, 10HXX XX, 10HXX XX XX,	HEADINX	56
C	6 2*10HXX XX XX, 10HXX XX XX, 10HXX XXXX, 10HXX XXX,	HEADINX	57
C	7 10HXX XX, 9H XXXXXXXX, 10HXXXXXXXXXXXX, 6*10HXX XX,	HEADINX	58



	8 10MXXXXXXXXXX, 9M XXXXXXXX /	HEADINX	59
C	DATA (KK(I), I = 151,280)	HEADINX	60
	\$ / 9MXXXXXXXXXX, 10MXXXXXXXXXX, 2*10MXX XX,	HEADINX	61
	1 10MXXXXXXXXXX, 9MXXXXXXXXXX, 4*2MXX, 9M XXXXXXXX, 10MXXXXXXXXXX,	HEADINX	62
	2 4*10MXX XX, 10MXX XX XX, 10MXX XXXX, 9MXXXXXXXXXX,	HEADINX	63
	3 10M XXXXX XX, 9MXXXXXXXXXX, 10MXXXXXXXXXX, 2*10MXX XX,	HEADINX	64
	4 10MXXXXXXXXXX, 9MXXXXXXXXXX, 7MXX XX, 8MXX XX, 9MXX XX,	HEADINX	65
	5 10MXX XX, 9M XXXXXXXX, 10MXXXXXXXXXX, 10MXX XX, 2MXX,	HEADINX	66
	6 9MXXXXXXXXXX, 10M XXXXXXXX, 10M XX, 10MXX XX,	HEADINX	67
	7 10MXXXXXXXXXX, 9M XXXXXXXX, 2*10MXXXXXXXXXX, 8*6M XX,	HEADINX	68
	8 8*10MXX XX, 10MXXXXXXXXXX, 9M XXXXXXXX, 2*10MXX XX,	HEADINX	69
	9 4*10M XX XX, 2*8M XX XX, 7M XXXX, 6M XX,	HEADINX	70
	\$ 4*10MXX XX, 3*10MXX XX XX, 10MXX XXXX XX, 10M XXX XXX,	HEADINX	71
	1 9M XX XX, 10MXX XX, 9M XX XX, 8M XX XX, 7M XXXX,	HEADINX	72
	2 6M XX, 7M XXXX, 8M XX XX, 9M XX XX, 3*10MXX XX,	HEADINX	73
	3 9M XX XX, 8M XX XX, 7M XXXX, 6*6M XX, 10MXXXXXXXXXX,	HEADINX	74
	4 9MXXXXXXXXXX, 8M XX, 7M XX, 2*6M XX, 5M XX, 4M XX,	HEADINX	75
	5 10M XXXXXXXX, 10MXXXXXXXXXX, 6M XX, 6M XXXX, 6M XX XX,	HEADINX	76
	6 5*6M XX, 2*8M XXXXXX, 9M XXXXXXXX, 10MXXXXXXXXXX,	HEADINX	77
	7 10MXX XX, 10M XXXX, 9M XXX, 7M XX, 5M XX,	HEADINX	78
	8 4M XXX, 2*10MXXXXXXXXXX /	HEADINX	79
		HEADINX	80
C	DATA (KK(I), I = 281,410)	HEADINX	81
	\$ / 10MXXXXXXXXXX, 9MXXXXXXXXXX, 8M XX, 7M XXX,	HEADINX	82
	1 7M XXXXX, 9M XXXX, 10M XXX, 10MXX XXX, 9MXXXXXXXXXX,	HEADINX	83
	2 8M XXXXXX, 8M XXX, 8M XXXX, 8M XX XX, 8M XX XX,	HEADINX	84
	3 8M XX XX, 2*10MXXXXXXXXXX, 3*8M XX, 2*10MXXXXXXXXXX,	HEADINX	85
	4 2*2MXX, 9MXXXXXXXXXX, 10MXXXXXXXXXX, 10M XX, 10MXX XX,	HEADINX	86
	5 10MXXXXXXXXXX, 2*9M XXXXXXXX, 10MXXXXXXXXXX, 10MXX XX, 2MXX,	HEADINX	87
	6 9MXXXXXXXXXX, 10MXXXXXXXXXX, 2*10MXX XX, 10MXXXXXXXXXX,	HEADINX	88
	7 9M XXXXXXXX, 2*10MXXXXXXXXXX, 9M XXX, 7M XXX, 6M XXX,	HEADINX	89
	8 5M XXX, 4M XXX, 3*3MXX, 9M XXXXXXXX, 10MXXXXXXXXXX,	HEADINX	90
	9 2*10MXX XX, 2*9M XXXXXXXX, 2*10MXX XX, 10MXXXXXXXXXX,	HEADINX	91
	\$ 2*9M XXXXXXXX, 10MXXXXXXXXXX, 2*10MXX XX, 10MXXXXXXXXXX,	HEADINX	92
	1 10M XXXXXXXX, 10M XX, 10MXX XX, 10MXXXXXXXXXX,	HEADINX	93
	2 9M XXXXXXXX, 7M XXXX, 9M XXXXXXXX, 9M XX XX, 4*10MXX XX,	HEADINX	94
	3 9M XX XX, 9M XXXXXXXX, 7M XXXX, 1M, 3*6M XX, 2*9M XXXXXXXX,	HEADINX	95
	4 3*6M XX, 5*1M, 2*9M XXXXXXXX, 5*1M, 9M XX, 8M XX,	HEADINX	96
	5 7M XX, 6M XX, 5M XX, 4M XX, 3M XX, 2MXX, 2*1M, 6M XX,	HEADINX	97
	6 9M XX XX XX, 8M XXXXXX, 2*7M XXXX, 8M XXXXXX, 9M XX XX XX,	HEADINX	98
	7 6M XX, 3*1M, 2*6M XX, 2*1M, 2*6M XX, 2*1M /	HEADINX	99
		HEADINX	100
C	DATA (KK(I), I = 411,500)	HEADINX	101
	\$ / 6M XX, 9M XXXXXXXX, 10MXX XX XX, 6MXX XX,	HEADINX	102
	1 9MXXXXXXXXXX, 10M XXXXXXXX, 10M XX XX, 10MXX XX XX,	HEADINX	103
	2 9MXXXXXXXXXX, 6M XX, 2*1M, 2*9M XXXXXXXX, 2*1M, 2*9M XXXXXXXX,	HEADINX	104
	3 2*1M, 7M XX, 7M XXX, 5M XX, 4*4M XX, 5M XX, 6M XXX,	HEADINX	105
	4 7M XX, 5M XX, 6M XXX, 7M XX, 4*8M XX, 7M XX,	HEADINX	106
	5 6M XXX, 5M XX, 8M XX, 7M XX, 6M XX, 5M XX,	HEADINX	107
	6 2*4M XX, 5M XX, 6M XX, 7M XX, 8M XX, 4M XX,	HEADINX	108
	7 5M XX, 6M XX, 7M XX, 2*8M XX, 7M XX, 6M XX,	HEADINX	109
	8 5M XX, 4M XX, 7M XXXX, 8*5M XX, 2*7M XXXX, 8*7M XX,	HEADINX	110
	9 7M XXXX, 10*1M /	HEADINX	111
		HEADINX	112
C	NSYMBOL = 50	HEADINX	113
	BLANK = 1M	HEADINX	114
		HEADINX	115



```

      L1 = L2 = 0
      DO 5 L = 1,10
      IF (MESSAGE(L).NE.BLANK) GO TO 6
5     L1 = L1+1
6     DO 7 I = 1,10
      L = 11-I
      IF (MESSAGE(L).NE.BLANK) GO TO 8
7     L2 = L2+1
8     LB = L1 + L2
      IF (LB.GE.10) L1 = LB = 0
      LIP1 = L1 + 1
      NK = 10 - LB
      LB = (13*LB)/2
      NKPL1 = NK + L1
      DECODE (10,102,JPAGE) KSKIP
102  FORMAT (1X,A2,7X)
      IF (KSKIP.EQ.BLANK) GO TO 9
      ENCODE (10,101,KONTROL) JPAGE
101  FORMAT (A3,7H(//)//1X)
      GO TO 10
9     ENCODE (10,103,KONTROL) JPAGE
103  FORMAT (A1,9H//1X )
10   DO 1 L = LIP1,NKPL1
      DO 2 I = 1,NSYMBOL
      IF (MESSAGE(L).EQ.LETTER(I)) GO TO 1
2     CONTINUE
      I = NSYMBOL
1     NUM(L) = I
      DO 3 LINE = 1,10
      DO 4 L = LIP1, NKPL1
      N = NUM(L)
4     IMAGE(L) = KAR(LINE*N)
      ENCODE (30,100,FORM) KONTROL, LB, NK
100  FORMAT (3H(1H,A10,I2,1HX,I2,9H(3X,A10)))
      WRITE (6,FORM) (IMAGE(K), K = LIP1, NKPL1)
3     KONTROL = 1H
      RETURN
      END

```

```

HEADINX 116
HEADINX 117
HEADINX 118
HEADINX 119
HEADINX 120
HEADINX 121
HEADINX 122
HEADINX 123
HEADINX 124
HEADINX 125
HEADINX 126
HEADINX 127
HEADINX 128
HEADINX 129
HEADINX 130
HEADINX 131
HEADINX 132
HEADINX 133
HEADINX 134
HEADINX 135
HEADINX 136
HEADINX 137
HEADINX 138
HEADINX 139
HEADINX 140
HEADINX 141
HEADINX 142
HEADINX 143
HEADINX 144
HEADINX 145
HEADINX 146
HEADINX 147
HEADINX 148
HEADINX 149
HEADINX 150
HEADINX 151
HEADINX 152
HEADINX 153

```

	PROGRAM ELECT (INPUT,OUTPUT,TAPE5,TAPE6=OUTPUT,TAPE7,TAPE8,TAPE9,	ELECT	2
	1 TAPE10=INPUT)	ELECT	3
C	-----	ELECT	4
C		ELECT	5
C		ELECT	6
C	THIS CODE PERFORMS A NUMERICAL SOLUTION OF THE BOLTZMANN TRANS-	ELECT	7
C	PORT EQUATION FOR THE ELECTRON ENERGY DISTRIBUTION IN A WEAKLY	ELECT	8
C	IONIZED PLASMA IN THE PRESENCE OF AN ELECTRIC FIELD. MOMENTUM	ELECT	9
C	TRANSFER (WITH RECOIL), INELASTIC BINARY E-NEUTRAL PROCESSES (WITH	ELECT	10
C	SUPERELASTIC COLLISIONS), AND ELECTRON-ELECTRON SCATTERING ARE ALL	ELECT	11
C	INCLUDED, WITH CROSS SECTIONS PROVIDED BY AN ARBITRARILY LARGE EX-	ELECT	12
C	TERNAL FILE OF DATA. INPUT PARAMETERS ARE GAS MIXTURE, TEMPERA-	ELECT	13
C	TURE, PRESSURE, EXCITED STATE POPULATION DENSITIES, AND A SEQUENCE	ELECT	14
C	OF F/N VALUES. OUTPUT CONSISTS OF TABLES AND PLOTS OF PLASMA PA-	ELECT	15
C	RAMETERS, POWER PARTITIONING, <VSIG> EXCITATION AND DE-EXCITATION	ELECT	16
C	METERS, POWER PARTITIONING, EXCITATION AND DE-EXCITATION <VSIG>	ELECT	17
C	RATES, ETC.	ELECT	18
C		ELECT	19
C	*****	ELECT	20
C	•	ELECT	21
C	•	ELECT	22
C	• THIS CODE WAS DEVELOPED BY •	ELECT	23
C	•	ELECT	24
C	• DR. WILLIAM B. LACINA •	ELECT	25
C	• NORTHROP RESEARCH AND TECHNOLOGY •	ELECT	26
C	• ONE RESEARCH PARK •	ELECT	27
C	• PALOS VERDES PENINSULA, CA 90274 •	ELECT	28
C	• TEL: (213) 377-4811, EXT. 322 •	ELECT	29
C	•	ELECT	30
C	•	ELECT	31
C	*****	ELECT	32
C		ELECT	33
C	COMPLETE DOCUMENTATION OF THE PRESENT ANALYSIS (INCLUDING DISCUS-	ELECT	34
C	SION OF THE MATHEMATICAL FORMULATION, TECHNIQUES OF NUMERICAL SO-	ELECT	35
C	LUTION, DESCRIPTION OF OF SUBROUTINES, AND INSTRUCTION FOR USAGE)	ELECT	36
C	IS AVAILABLE IN PUBLISHED REPORTS BY W. B. LACINA. A COMPREHEN-	ELECT	37
C	SIVE AND GENERAL LASER KINETICS CODE IS ALSO AVAILABLE, AND MAKES	ELECT	38
C	USE OF THE SAME SUBROUTINES FOR THE ELECTRON KINETICS ANALYSIS.	ELECT	39
C	-----	ELECT	40
C		ELECT	41
C		ELECT	42
C	DIMENSION Q(1001,30), QM(1001), QMOM(1001,2), A(1001,3), F(1001),	ELECT	43
	1 EV(1001), G(1001), B(1001), POWER(30), N1(30), N2(30), P(21,31),	ELECT	44
	2 U(30), NEL(30), RATE(21,2,30), VSIG(2,30), PROCESS(4,31), FI(5),	ELECT	45
	3 IDENT(5), NAME(5), MASS(5), GAS(100), NO(100), E(100), TITLE(3),	ELECT	46
	4 Y0(2), DY(2), TABLE(21,9), FORM(15), KODE(8), HEAD(9), IMAGE(60),	ELECT	47
	5 EOVERN(21), EN(21), IOUT(10), OUT(10), KAPT(5), LABEL(5,2),	ELECT	48
	6 KINETIC(6,50), NUMBER(30), GMOLE(100), LINE(250), S(1001)	ELECT	49
C		ELECT	50
C	REAL NO, KB, KTE, MU, MASS, NMOL, N1, N2, NE, MOLWT, IONIZE	ELECT	51
C		ELECT	52
C	INTEGER GAS, TYPE, TITLE, TODAY, LHS, RHS	ELECT	53
C		ELECT	54
C	LOGICAL CONVRGE, OUT, FATAL, LIST, ERROR, STOP, TEST, MISSING,	ELECT	55
	1 OUTSIDE, REJECT, EXPAND, SEARCH, LIBRARY	ELECT	56
C		ELECT	57
	EQUIVALENCE (A,QM)	ELECT	58



C		ELECT	59
C		ELECT	60
C		ELECT	61
C	NAMelist / CONTROL / MESH, IOUT, FATAL, EMAX, ITMAX, TMAX, EPS,	ELECT	62
C	1 TE, IDEG, PCT, SEARCH	ELECT	63
C		ELECT	64
C	NAMelist / PARAM / TMOL, PTOT, ATM, EUVERN	ELECT	65
C		ELECT	66
C	NAMelist / SOURCE / DNEOT, BEAM, CREATE, UA, UB, S	ELECT	67
C		ELECT	68
C		ELECT	69
C		ELECT	70
C	DATA KR, EE / 1.38E-23, 1.602E-19 /	ELECT	71
C		ELECT	72
C	DATA EN / .1, .2, .3, .5, .8, 1., 2., 3., 5., 12*0. /	ELECT	73
C		ELECT	74
C	DATA HEAD / 6HE/NTOT, 10H<U>= 3TE/2, 9HEK = D/MU, 10HTE (DEG K),	ELECT	75
C	1 9HVD (CM/S), 10HMU, CM2/V/S, 9HD, CM2/S, 9HP/NE/NMOL,	ELECT	76
C	2 9HJ/NE=E*VD /	ELECT	77
C		ELECT	78
C	DATA TITLE, KAPT / 8HELECTRON, 8MKINETICS, 8HANALYSIS, 5*IH /	ELECT	79
C		ELECT	80
C		ELECT	81
C		ELECT	82
C	DATA INPUT SEQUENCE --	ELECT	83
C		ELECT	84
C		ELECT	85
C		ELECT	86
C	IOUT(10) IS A VECTOR OF OUTPUT OPTIONS -- OPTION 1 IS SUP-	ELECT	87
C	PRESSED IF IOUT(1) = 0, OTHERWISE PROVIDED --	ELECT	88
C		ELECT	89
C	IOUT(1) -- INDIVIDUAL SUMMARY OF PLASMA PARAMETERS FOR	ELECT	90
C	EACH E/N VALUE	ELECT	91
C	IOUT(2) -- TABLE OF ELECTRON DISTRIBUTION FUNCTION FOR	ELECT	92
C	EACH VALUE OF E/N	ELECT	93
C	IOUT(3) -- PLOT OF ELECTRON DISTRIBUTION FUNCTION FOR	ELECT	94
C	EACH VALUE OF E/N	ELECT	95
C	IOUT(4) -- PLOT OF F(U)/FBOLTZ(U,TE) FOR EACH E/N	ELECT	96
C	IOUT(5) -- SUMMARY OF PLASMA PARAMETERS, COLLISION	ELECT	97
C	RATES, AND POWER BALANCE FOR EACH E/N	ELECT	98
C	IOUT(6) -- TABULAR SUMMARY OF INPUT CROSS SECTION DATA	ELECT	99
C	IOUT(7) -- PLOTS OF INPUT CROSS SECTIONS (IOUT(6) MUST	ELECT	100
C	BE SIMULTANEOUSLY SPECIFIED)	ELECT	101
C	IOUT(8) -- TABULAR SUMMARY OF PLASMA PARAMETERS AS A	ELECT	102
C	FUNCTION OF E/N FOR THE GIVEN GAS MIXTURE;	ELECT	103
C	TABLES OF POWER PARTITIONING AS A FUNCTION OF	ELECT	104
C	E/N FOR ELASTIC AND INELASTIC PROCESSES.	ELECT	105
C	IOUT(9) -- PLOTS OF PLASMA PARAMETERS AS A FUNCTION OF	ELECT	106
C	E/N FOR A GIVEN GAS MIXTURE (IF AT LEAST 5	ELECT	107
C	CASES HAVE BEEN COMPUTED)	ELECT	108
C	IOUT(10) -- LOG PLOTS OF FORWARD (AND REVERSE) ELECTRON	ELECT	109
C	RATES AS A FUNCTION OF E/N, FOR EACH OF THE	ELECT	110
C	INELASTIC COLLISION PROCESSES INCLUDED	ELECT	111
C		ELECT	112
C	MISCELLANEOUS PARAMETERS DEFINED --	ELECT	113
C		ELECT	114
C	EMAX -- MAXIMUM ELECTRON ENERGY (EV)	ELECT	115
C	MESH -- NUMBER OF INTERVALS INTO WHICH THE ENERGY RANGE		



C		IS DIVIDED. (MESH $\leq 1000$ IS INSURED BY PROGRAM.)	ELECT	116
C	EVCM	-- ELECTRIC FIELD (VOLT/CM)	ELECT	117
C	EOVERN	-- E/N ( $1.E-16$ VOLT CM <sup>2</sup> )	ELECT	118
C	TE	-- INITIAL GUESS FOR ELECTRON TEMPERATURE (DEG K)	ELECT	119
C	ITMAX	-- MAXIMUM NUMBER OF ITERATIONS PERMITTED.	ELECT	120
C	TMAX	-- MAXIMUM CP TIME FOR OBTAINING CONVERGENCE.	ELECT	121
C	EPS	-- CONVERGENCE CRITERION FOR MAXIMUM RELATIVE	ELECT	122
C		CHANGE FOR ALL VALUES OF THE ELECTRON DISTRIBUTION FUNCTION BETWEEN SUCCESSIVE ITERATIONS.	ELECT	123
C	IDEG	-- DEGREE OF INTERPOLATION FOR CROSS SECTIONS TO	ELECT	124
C		GENERATE UNIFORM GRID OF VALUES FROM RAW DATA.	ELECT	125
C			ELECT	126
C	INPUT CONSISTS OF THE FOLLOWING --		ELECT	127
C			ELECT	128
C			ELECT	129
C	A) DECK OF ELECTRON CROSS SECTION PACKAGES (ARBITRARILY MANY)		ELECT	130
C	FOR UPDATE OF ELECTRON CROSS SECTION DATA FILE, TERMINATED		ELECT	131
C	BY AN EOF (7/8/9) CARD.		ELECT	132
C			ELECT	133
C	B) LIST OF (UP TO LIMIT) ELECTRON KINETIC REACTIONS, TERMINATED BY AN EOF (7/8/9) CARD. THESE REACTIONS FORM THE BASIS FOR ALL SUBSEQUENT ELECTRON KINETICS CALCULATIONS. HOWEVER, IF NO REACTIONS ARE ENTERED IN THIS FILE, THE PROGRAM WILL SEARCH THE EXTERNAL ELECTRON CROSS SECTION FILE AND RETAIL ALL RELEVANT REACTIONS (I.E. WHICH CONTAIN GASES IN THE SPECIFIED MIXTURE). REGARDLESS OF WHERE THE KINETIC SCHEME IS FOUND, ONLY THE FIRST NKMAX LEGAL REACTIONS ARE RETAINED.		ELECT	134
C			ELECT	135
C			ELECT	136
C			ELECT	137
C			ELECT	138
C			ELECT	139
C			ELECT	140
C			ELECT	141
C			ELECT	142
C			ELECT	143
C	THESE ARE FOLLOWED BY ARBITRARILY MANY OF THE FOLLOWING PACKAGES, EACH OF WHICH IS TERMINATED BY AN EOF (7/8/9) CARD --		ELECT	144
C			ELECT	145
C			ELECT	146
C	1) SCUNCONTROL ... S		ELECT	147
C	2) S PARAM ... S		ELECT	148
C	3) PACKAGE OF SPECIES CARDS, CONTAINING: NAME, FRACTION P0, ENERGY (EV), MOLECULAR WEIGHT (GM/MOLE). (A10.3E10.3)		ELECT	149
C			ELECT	150
C			ELECT	151
C	THE FRACTIONAL GAS COMPOSITION IS DEFINED BY THE VALUES ENTERED FOR P0 ON THE SPECIES CARDS. IF THE TOTAL OF THESE VALUES IS MORE THAN 1.0E 08, IT IS ASSUMED THAT THE P0 VALUES REPRESENT POPULATION DENSITIES (CM <sup>-3</sup> ), AND ANY VALUES ENTERED FOR PTOT OR ATM ARE IGNORED. IF THE TOTAL IS LESS THAN 1.0E 08, IT IS ASSUMED THAT THE P0 VALUES REPRESENT PARTIAL PRESSURES (IN TORR), UNLESS EITHER PTOT OR ATM IS SPECIFIED ON THE SPARAM ...S CARD, IN WHICH CASE THEY REPRESENT ONLY FRACTIONAL CONCENTRATIONS. IF BOTH PTOT (IN TORR) AND ATM (IN ATMOSPHERES) IS SPECIFIED, THE VALUE FOR PTOT IS ACCEPTED, AND ATM IS IGNORED.		ELECT	152
C			ELECT	153
C			ELECT	154
C			ELECT	155
C			ELECT	156
C			ELECT	157
C			ELECT	158
C			ELECT	159
C			ELECT	160
C			ELECT	161
C			ELECT	162
C	-----		ELECT	163
C	CALL DATE (TODAY)		ELECT	164
C			ELECT	165
C	DIMENSION DECLARATORS --		ELECT	166
C	MGRID = 1000		ELECT	167
	NMAX = 5		ELECT	168
	NKMAX = 30		ELECT	169
	LIMIT = 50		ELECT	170
	MAX = 100		ELECT	171
			ELECT	172

C		ELECT	173
C	SCRATCH FILE FOR E- UPDATE --	ELECT	174
	NTAPE = 7	ELECT	175
C		ELECT	176
C	INPUT E- CROSS SECTION TAPE FILE --	ELECT	177
	INPUT = 8	ELECT	178
	REWIND INPUT	ELECT	179
C		ELECT	180
C	FILE OF UPDATED E- CROSS SECTIONS --	ELECT	181
	NDATA = 9	ELECT	182
C		ELECT	183
C	INPUT CARD FILE --	ELECT	184
	KARDS = 10	ELECT	185
C		ELECT	186
	KB = KB/EE	ELECT	187
	LIST = .TRUE.	ELECT	188
	CALL EDITOR (KARDS, LIST)	ELECT	189
	CALL UPDATE (INPUT, NDATA, NTAPE, .FALSE., TODAY)	ELECT	190
	CALL COVER (TITLE,2)	ELECT	191
C		ELECT	192
C	-----	ELECT	193
C	ENTER AN INITIAL REACTION SCHEME, IF DESIRED --	ELECT	194
C	-----	ELECT	195
C		ELECT	196
C	UNLESS SEARCH = TRUE IS SPECIFIED ON SCONTROL...S CARD BELOW, IT	ELECT	197
C	IS ASSUMED THAT USER WISHES TO RESTRICT CALCULATIONS TO NO MORE	ELECT	198
C	THAN THE FOLLOWING PROCESSES. IF NONE ARE ENTERED, PROGRAM WILL	ELECT	199
C	AUTOMATICALLY SEARCH THE EXTERNAL ELECTRON CROSS SECTION FILE AND	ELECT	200
C	USE ALL RELEVANT KINETIC PROCESSES INVOLVING SPECIES FOUND IN THE	ELECT	201
C	MIXTURE SPECIFIED.	ELECT	202
C		ELECT	203
	CALL EDITOR (KARDS,LIST)	ELECT	204
	DO 28 N = 1,LIMIT	ELECT	205
	DO 29 I = 1,6	ELECT	206
28	KINETIC(I,N) = 1H	ELECT	207
	N = 0	ELECT	208
	LIBRARY = .TRUE.	ELECT	209
53	READ (5,250) CODE	ELECT	210
	IF (EOF(5)) 50,33	ELECT	211
33	IF (N.EQ.LIMIT) GO TO 53	ELECT	212
	N = N+1	ELECT	213
	LIBRARY = .FALSE.	ELECT	214
	DO 51 I = 1,6	ELECT	215
51	KINETIC(I,N) = CODE(I)	ELECT	216
	GO TO 53	ELECT	217
C	ALL KINETIC REACTIONS HAVE BEEN STORED.	ELECT	218
C		ELECT	219
C	-----	ELECT	220
C	READ IN PACKAGE OF INPUT PARAMETERS FOR EXECUTION OF ANALYSIS --	ELECT	221
C	-----	ELECT	222
C		ELECT	223
50	CALL EDITOR (KARDS,LIST)	ELECT	224
C		ELECT	225
C	ENTER CALCULATION CONTROL PARAMETERS --	ELECT	226
C		ELECT	227
	MESH = 1000	ELECT	228
	E <sub>MAX</sub> = 5.0	ELECT	229



EXPAND = .FALSE.	ELECT	230
PCT = 10.	ELECT	231
PERCENT = PCT * 1.	ELECT	232
TMAX = 50.	ELECT	233
ITMAX = 100	ELECT	234
EPS = .001	ELECT	235
IUEG = 2	ELECT	236
NPTS = 21	ELECT	237
DO 3 I = 1,10	ELECT	238
3 IOUT(I) = 1	ELECT	239
DO 31 N = 1,NMAX	ELECT	240
NAME(N) = 1H	ELECT	241
IDENT(N) = 0	ELECT	242
31 FI(N) = MASS(N) = 0.	ELECT	243
DO 13 N = 1,MAX	ELECT	244
GMOLE(N) = 0.	ELECT	245
GAS(N) = 1H	ELECT	246
13 E(N) = NO(N) = 0.	ELECT	247
DO 18 N = 1,NPTS	ELECT	248
18 EUVERN(N) = -1.0	ELECT	249
TE = 0.	ELECT	250
SEARCH = FATAL = .FALSE.	ELECT	251
C	ELECT	252
C	ELECT	253
-----	ELECT	254
READ (5,CONTROL)	ELECT	255
C	ELECT	256
C	ELECT	257
ENTER EXPERIMENTAL CONDITIONS --	ELECT	258
IF (EOF(5)) 99,72	ELECT	259
72 DO 42 I = 1,10	ELECT	260
42 OUT(I) = IOUT(I).NE.0	ELECT	261
TMOL = 300.	ELECT	262
PTOT = ATM = 0.	ELECT	263
C	ELECT	264
C	ELECT	265
-----	ELECT	266
READ (5,PARAM)	ELECT	267
C	ELECT	268
IF (EOF(5)) 99,71	ELECT	269
71 IF (PTOT.EQ.0.) PTOT = 760.*ATM	ELECT	270
IF (TMOL.LE.0.) TMOL = 300.	ELECT	271
IF (TE.EQ.0.) TE = TMOL	ELECT	272
KTE = KB*TE	ELECT	273
IF (MESH.GT.MGRID) MESH = MGRID	ELECT	274
M = MESH * 1	ELECT	275
C	ELECT	276
SEARCH = SEARCH.OR.LIBRARY	ELECT	277
C	ELECT	278
NE = IONIZE = 0.	ELECT	279
NTYPE = 0	ELECT	280
C	ELECT	281
ENTER EXTERNAL DEPOSITION SOURCE DATA --	ELECT	282
DNEDT = BEAM = CREATE = 0.	ELECT	283
UA = UB = 0.	ELECT	284
C	ELECT	285
C	ELECT	286
-----		
READ (5,SOURCE)		
C		



C	IF (EOF(5)) 99,73	ELECT	287
	73 IF (UB.GT.EMAX) UB = EMAX	ELECT	288
	UA = UA*1.000001	ELECT	289
	UB = UB*1.000001	ELECT	290
C		ELECT	291
C	ENTER SPECIES PARAMETERS --	ELECT	292
C		ELECT	293
C	-----	ELECT	294
C	20 READ (5,110) TYPE, P0, E0, MOLWT	ELECT	295
C	-----	ELECT	296
C		ELECT	297
C	IF (EOF(5)) 11,6	ELECT	298
C	REJECT MORE THAN MAX DIFFERENT SPECIES --	ELECT	299
	6 IF (NTYPE.EQ.MAX) GF TO 20	ELECT	300
C	PROGRAM USES THE LATEST VALUES READ FOR E(-) OR IONIZE --	ELECT	301
	IF (TYPE.NE.4HE(-).AND.TYPE.NE.2HE-.AND.TYPE.NE.1HE) GO TO 34	ELECT	302
	NE = P0	ELECT	303
	GO TO 20	ELECT	304
	34 IF (TYPE.NE.6HIONIZE) GO TO 39	ELECT	305
	IONIZE = P0	ELECT	306
	GO TO 20	ELECT	307
C	REJECT DUPLICATION OF SPECIES --	ELECT	308
	39 IF (NTYPE.EQ.0) GO TO 64	ELECT	309
	DO 62 I = 1,NTYPE	ELECT	310
	IF (TYPE.NE.GAS(I)) GO TO 62	ELECT	311
C	PREVIOUS VALUES FOR GAS(I) ARE REPLACED BY MOST RECENT --	ELECT	312
	E(I) = E0	ELECT	313
	NO(I) = P0	ELECT	314
	GMOLE(I) = MOLWT	ELECT	315
	GO TO 20	ELECT	316
	62 CONTINUE	ELECT	317
C		ELECT	318
C	STORE PARAMETERS FOR THE NEW SPECIES CALLED *TYPE* --	ELECT	319
	64 NTYPE = NTYPE+1	ELECT	320
	GAS(NTYPE) = TYPE	ELECT	321
	E(NTYPE) = E0	ELECT	322
	NO(NTYPE) = P0	ELECT	323
	GMOLE(NTYPE) = MOLWT	ELECT	324
	GO TO 20	ELECT	325
C		ELECT	326
C	PARAMETERS USED FOR MOMENTUM TRANSFER --	ELECT	327
C		ELECT	328
	11 PRESS = NMOL = NGAS = 0	ELECT	329
	DO 69 I = 1,NTYPE	ELECT	330
	69 PRESS = PRESS + NO(I)	ELECT	331
C	SORT SPECIES IN DESCENDING ORDER OF COMPOSITION --	ELECT	332
	68 PMAX = 0.	ELECT	333
	DO 1 I = 1,NTYPE	ELECT	334
	DO 65 K = 1,NGAS	ELECT	335
	IF (I.EQ.IDENT(K)) GO TO 1	ELECT	336
	65 CONTINUE	ELECT	337
	IF (NO(I).LE.PMAX) GO TO 1	ELECT	338
	PMAX = NO(I)	ELECT	339
	NG = I	ELECT	340
	1 CONTINUE	ELECT	341
C	REJECT COMPONENTS < 0.01 FROM MOMENTUM TRANSFER CALCULATIONS --	ELECT	342
		ELECT	343

TEST = (NGAS.NF 1).AND.(PMAx.LT.1.0E-04*FI(1))	ELECT	344
IF (TEST) GO TO 66	ELECT	345
NGAS = NGAS + 1	ELECT	346
IDENr(NGAS) = NG	ELECT	347
NAME(NGAS) = GAS(NG)	ELECT	348
MASS(NGAS) = GMOLE(NG)	ELECT	349
FI(NGAS) = NO(NG)	ELECT	350
IF (NGAS.LT.NMAX) GO TO 68	ELECT	351
C	ELECT	352
66 NMOL = PRESS	ELECT	353
DO 67 I = 1,NGAS	ELECT	354
67 FI(I) = FI(I)/PRESS	ELECT	355
IF (NMOL.GT.1.0E 08) GO TO 54	ELECT	356
IF (PTOT.EQ.0.) PTOT = PRESS	ELECT	357
NMOL = 0.965E 19*PTOT/TMOL	ELECT	358
FRACT = NMOL/PRESS	ELECT	359
DO 9 I = 1,NTYPE	ELECT	360
9 NO(I) = NO(I)*FRACT	ELECT	361
54 IF (NE.LE.0.) NE = IONIZE*NMOL	ELECT	362
NP1 = NTYPE+1	ELECT	363
GAS(NP1) = 4HE(-)	ELECT	364
E(NP1) = 0.	ELECT	365
KOUNT = NP1	ELECT	366
DO 7 N = 1,NKMAX	ELECT	367
7 U(N) = N1(N) = N2(N) = 0.	ELECT	368
C	ELECT	369
N = NK = 0	ELECT	370
C	ELECT	371
C	ELECT	372
C-----	ELECT	372
C SELECT, FROM THE INPUT REACTION QUEUE, ALL (LEGAL) REACTIONS WHICH	ELECT	373
C INVOLVE SPECIES WHICH HAVE BEEN ENTERED --	ELECT	374
C-----	ELECT	375
C	ELECT	376
35 IF (SEARCH) GO TO 70	ELECT	377
IF (N.EQ.LIMIT) GO TO 5	ELECT	378
N = N+1	ELECT	379
DECODE (60,260,KINETIC(1,N)) IMAGE	ELECT	380
GO TO 61	ELECT	381
C	ELECT	382
C-----	ELECT	383
C SEARCH THE EXTERNAL ELECTRON CROSS SECTION FILE AND RETAIN ALL	ELECT	384
C RELEVANT REACTIONS (UP TO NKMAX) FOR THE INPUT GAS MIXTURE --	ELECT	385
C-----	ELECT	386
C	ELECT	387
70 READ (NDATA,260) (IMAGE(L), L = 1,60)	ELECT	388
IF (EOF(NDATA)) 5,63	ELECT	389
63 N = NK+1	ELECT	390
C	ELECT	391
C DUMP NUMERICAL DATA --	ELECT	392
READ (NDATA)	ELECT	393
59 READ (NDATA,250) KODE	ELECT	394
IF (KODE(1).EQ.1H .AND.KODE(2).EQ.1H ) GO TO 61	ELECT	395
GO TO 59	ELECT	396
C	ELECT	397
61 IF (NK.EQ.NKMAX) GO TO 5	ELECT	398
CALL DEKODE (GAS, IMAGE, LHS, RHS, LABEL, IOUT, 10, KOUNT, 60)	ELECT	399
C RECALL THAT KOUNT IS AUTOMATICALLY UPDATED UPON RETURN FROM DEKODE	ELECT	400



C		ELECT	401
C	ELIMINATE MOMENTUM TRANSFER PROCESSES --	ELECT	402
	IF (LHS.EQ.RHS) GO TO 35	ELECT	403
C		ELECT	404
C	IF REACTION CONTAINS NONE OF THE SPECIES ENTERED, REJECT IT --	ELECT	405
C		ELECT	406
	DO 55 L = 1,5	ELECT	407
	I = LABEL(L,1)	ELECT	408
	IF (I.LE.0) GO TO 58	ELECT	409
	IF (I.LE.NTYPE) GO TO 56	ELECT	410
58	I = LABEL(L,2)	ELECT	411
	IF (I.LE.0) GO TO 55	ELECT	412
	IF (I.LE.NTYPE) GO TO 56	ELECT	413
55	CONTINUE	ELECT	414
	GO TO 35	ELECT	415
C		ELECT	416
56	NL = NR = 0	ELECT	417
	UK = 0.	ELECT	418
	L1 = L2 = 0	ELECT	419
	LEVEL = 0	ELECT	420
	DO 48 L = 1,5	ELECT	421
	I = LABEL(L,1)	ELECT	422
	IF (I.EQ.0) GO TO 43	ELECT	423
	UK = UK - E(I)	ELECT	424
	IF (I.NE.NP1) LEVEL = I	ELECT	425
48	IF (I.EQ.NP1) NL = NL + 1	ELECT	426
43	IF (NL.EQ.1.AND.L.EQ.3) L1 = LEVEL	ELECT	427
	LEVEL = 0	ELECT	428
	DO 49 L = 1,5	ELECT	429
	I = LABEL(L,2)	ELECT	430
	IF (I.EQ.0) GO TO 38	ELECT	431
	UK = UK + E(I)	ELECT	432
	IF (I.NE.NP1) LEVEL = I	ELECT	433
49	IF (I.EQ.NP1) NR = NR + 1	ELECT	434
38	IF (NR.EQ.1.AND.L.EQ.3) L2 = LEVEL	ELECT	435
	IF (L1.L2.EQ.0) GO TO 35	ELECT	436
C		ELECT	437
	NK = NK+1	ELECT	438
	IF (L1.NE.0) N1(NK) = NO(L1)	ELECT	439
	IF (L2.NE.0) N2(NK) = NO(L2)	ELECT	440
	IF (N1(NK)*N2(NK).NE.0.) GO TO 47	ELECT	441
	NK = NK-1	ELECT	442
	GO TO 35	ELECT	443
47	U(NK) = UK	ELECT	444
	NEL(NK) = NR - NL	ELECT	445
	ENCODE (40,260,PROCESS(1,NK)) (IMAGE(L), L = 1,40)	ELECT	446
	NUMBER(NK) = N	ELECT	447
	ENCODE (60,260,KINETIC(1,N)) (IMAGE(L), L = 1,60)	ELECT	448
	GO TO 35	ELECT	449
C		ELECT	450
5	NKP1 = NK+1	ELECT	451
	REWIND NDATA	ELECT	452
	ENCODE (40,320,PROCESS(1,NKP1))	ELECT	453
C		ELECT	454
	IF (.NOT.EXPAND) GO TO 22	ELECT	455
C		ELECT	456
C	-----	ELECT	457



C	IF THE BOLTZMANN ANALYSIS BELOW DID NOT CONVERGE, CONTROL RETURNS	ELECT	458
C	TO THIS POINT TO EXPAND THE ELECTRON ENERGY RANGE --	ELECT	459
C	-----	ELECT	460
C		ELECT	461
	40 IF (EMAX.GE.5.) GO TO 23	ELECT	462
	EMAX = EMAX + EMAX	ELECT	463
	IF (EMAX.GT.5.) EMAX = 5.	ELECT	464
	GO TO 27	ELECT	465
	23 EMAX = EMAX + 5.0	ELECT	466
	27 IF (EMAX.GT.25.) GO TO 50	ELECT	467
C		ELECT	468
	22 E0 = 0.	ELECT	469
	DE = EMAX/MESH	ELECT	470
	NA = UA/DE+1	ELECT	471
	NB = UB/DE+1	ELECT	472
	IF ((UA.NE.UB).AND.(NA.EQ.NB)) NB = NA+1	ELECT	473
	DU = (NB-NA)*DE	ELECT	474
	SI = 0.	ELECT	475
	IF (DU.GT.0.) SI = 1./DU	ELECT	476
	DO 29 I = 1,M	ELECT	477
	QMOM(I,1) = QMOM(I,2) = 0.	ELECT	478
	EV(I) = E0	ELECT	479
	E0 = E0 + DE	ELECT	480
	S(I) = 0.	ELECT	481
	IF (I.LT.NA.OR.I.GT.NB) GO TO 29	ELECT	482
	S(I) = SI	ELECT	483
	29 QM(I) = EV(I)*S(I)	ELECT	484
	BEAM = BEAM*DU*SI	ELECT	485
C		ELECT	486
C	PLOT SOURCE FUNCTION S(U) FOR EXTERNAL DEPOSITION --	ELECT	487
	IF (BEAM.EQ.0.) GO TO 45	ELECT	488
	CALL SIMPSON (QM, MESH/2, DE, DEPOSIT)	ELECT	489
	DEPOSIT = 1.602E-19*BEAM*DEPOSIT	ELECT	490
	WRITE (6,360) BEAM, DEPOSIT	ELECT	491
	Y0(1) = DY(1) = 0.	ELECT	492
	CALL PLOT (1, M, 1, S, Y0, DY, EV, 0., 0., .TRUE., .TRUE., .TRUE.,	ELECT	493
	1 .TRUE., .TRUE., TITLE, 1, 0)	ELECT	494
C		ELECT	495
	WRITE (6,370) TODAY	ELECT	496
C		ELECT	497
C	-----	ELECT	498
C	PROCESS THE ELECTRON CROSS SECTION ARRAYS --	ELECT	499
C	-----	ELECT	500
C		ELECT	501
	45 K = 0	ELECT	502
	J = 1	ELECT	503
	ERROR = .FALSE.	ELECT	504
	DO 57 N = 1,NK	ELECT	505
	I = NUMBER(N)	ELECT	506
	DECODE (60,260,KINETIC(1,I)) IMAGE	ELECT	507
	CALL DECODE (GAS, IMAGE, LHS, RHS, LABEL, IOUT, 10, KOUNT, 60)	ELECT	508
C		ELECT	509
C	PROCESS THE NTH INELASTIC ELECTRON CROSS SECTION --	ELECT	510
C		ELECT	511
	CALL PLASMA (NDATA, MGRID+1, MESH, LHS, RHS, PROCESS, EV, F, G,	ELECT	512
	1 Q(1,N), UTH, UM, KOUNT, GAS, MISSING, REJECT, OUTSIDE, IDEG,	ELECT	513
	2 OUT(6))	ELECT	514

C	STOP = MISSING.OR.REJECT.OR.OUTSIDE	ELECT	515
	IF (.NOT.STOP) GO TO 32	ELECT	516
	IF (J.GT.231) GO TO 57	ELECT	517
	ENCODE (50,340,LINE(J))	ELECT	518
	J = J+5	ELECT	519
	K = K+1	ELECT	520
	ENCODE (50,111,LINE(J)) K, (PROCESS(L,N), L = 1,4)	ELECT	521
	J = J+5	ELECT	522
	IF (MISSING) ENCODE (50,115,LINE(J))	ELECT	523
	IF (MISSING) J = J+5	ELECT	524
	IF (OUTSIDE) ENCODE (50,116,LINE(J)) EMAX	ELECT	525
	IF (OUTSIDE) J = J+5	ELECT	526
	IF (REJECT) ENCODE (50,117,LINE(J)) EMAX	ELECT	527
	IF (REJECT) J = J+5	ELECT	528
	IF (J.LE.225) GO TO 57	ELECT	529
	ENCODE (100,330,LINE(J))	ELECT	530
	J = J+10	ELECT	531
	GO TO 57	ELECT	532
C	32 DO 24 L = 1,M	ELECT	533
	24 Q(L,N) = EV(L)*Q(L,N)	ELECT	534
	57 ERROR = ERROR.OR.STOP	ELECT	535
C	PROCESS THE MOMENTUM TRANSFER CROSS SECTIONS --	ELECT	536
C	TWOM = 2./1837.	ELECT	537
	DO 17 I = 1,NGAS	ELECT	538
	MISSING = REJECT = OUTSIDE = .FALSE.	ELECT	539
	ENCODE (40,113,KAPT) NAME(I)	ELECT	540
	ID = IDENT(I)	ELECT	541
	ID1 = ID + NP1	ELECT	542
	ID2 = ID + NP1*NP1	ELECT	543
	ENCODE (10,114,LHS) ID1, ID2	ELECT	544
	RHS = LHS	ELECT	545
C	CALL PLASMA (NDATA, MGRID+1, MESH, LHS, RHS, KAPT, EV, F, G, OM,	ELECT	546
	1 UTH, UM, KOUNT, GAS, MISSING, REJECT, OUTSIDE, IDEG, OUT(6))	ELECT	547
C	STOP = MISSING.OR.REJECT.OR.OUTSIDE.OR.(MASS(I).LE.0.).OR.	ELECT	548
	(UM.LT.EMAX)	ELECT	549
1	IF (.NOT.STOP) GO TO 36	ELECT	550
	IF (J.GT.221) GO TO 17	ELECT	551
	ENCODE (50,340,LINE(J))	ELECT	552
	J = J+5	ELECT	553
	K = K+1	ELECT	554
	ENCODE (50,111,LINE(J)) K, (KAPT(L), L = 1,4)	ELECT	555
	J = J+5	ELECT	556
	IF (MISSING) ENCODE (50,115,LINE(J))	ELECT	557
	IF (MISSING) J = J+5	ELECT	558
	IF (OUTSIDE) ENCODE (50,116,LINE(J)) EMAX	ELECT	559
	IF (OUTSIDE) J = J+5	ELECT	560
	IF (REJECT) ENCODE (50,117,LINE(J)) EMAX	ELECT	561
	IF (REJECT) J = J+5	ELECT	562
	IF (UM.LT.EMAX) ENCODE (50,118,LINE(J)) EMAX	ELECT	563
	IF (UM.LT.EMAX) J = J+5	ELECT	564
	IF (MASS(I).LE.0.) ENCODE (50,119,LINE(J))	ELECT	565
		ELECT	566
		ELECT	567
		ELECT	568
		ELECT	569
		ELECT	570
		ELECT	571



```

      IF (MASS(I).LE.0.) J = J+5
      IF (J.LE.225) GO TO 17
      ENCODE (100,330,LINE(J))
      J = J+10
      GO TO 17

C-----
C  CONSTRUCT TWO MOMENTUM TRANSFER FUNCTIONS WHICH OCCUR IN THE
C  BOLTZMANN EQUATION --
C-----

36  FRACT = FI(I)
    DO 44 L = 1,M
      FQ = FRACT*QM(L)
      QMOM(L,1) = QMOM(L,1) + FQ
44  QMOM(L,2) = QMOM(L,2) + FQ/MASS(I)

C  17  ERROR = ERROR.OR.STOP

C-----
C  GENERATE SUMMARY OF WARNING DIAGNOSTICS, IF ANY --
C-----

      J = J-1
      IF (J.GT.0) WRITE (6,112) (LINE(L), L = 1,J)

C
      DO 46 L = 1,M
        X = EV(L)
        XSQ = TWOM*X*X
        STOP = QMOM(L,1).EQ.0.
        IF (STOP) GO TO 8
        A(L,1) = X/NMOL/QMOM(L,1)
46  A(L,2) = XSQ/NMOL*QMOM(L,2)

C
      XBAR = DE/2.
      DO 41 I = 1,MESH
        CALL INTERP (2, XBAR, QMOM(I,1), EV, A(I,1), 1, M)
        CALL INTERP (2, XBAR, QMOM(I,2), EV, A(I,2), 1, M)
41  XBAR = XBAR + DE
      QMOM(M,1) = QMOM(MESH,1)
      QMOM(M,2) = QMOM(MESH,2)

C
      8  ERROR = ERROR.OR.STOP
      FATAL = FATAL.AND.ERROR
      IF (STOP) WRITE (6,350)
      IF (FATAL) GO TO 50
      EXPON = EXP(-DE/KTE)
      FB = 1.
      DO 37 I = 1,M
        FI(I) = FB
37  FB = FB*EXPON

C
      ENCODE (40,500,KAPT)

C
C  PHOIBIT FURTHER PLOTS OR TABULATIONS OF E- CROSS SECTION DATA--
C
      OUT(6) = OUT(7) = .FALSE.

```

```

ELECT 572
ELECT 573
ELECT 574
ELECT 575
ELECT 576
ELECT 577
ELECT 578
ELECT 579
ELECT 580
ELECT 581
ELECT 582
ELECT 583
ELECT 584
ELECT 585
ELECT 586
ELECT 587
ELECT 588
ELECT 589
ELECT 590
ELECT 591
ELECT 592
ELECT 593
ELECT 594
ELECT 595
ELECT 596
ELECT 597
ELECT 598
ELECT 599
ELECT 600
ELECT 601
ELECT 602
ELECT 603
ELECT 604
ELECT 605
ELECT 606
ELECT 607
ELECT 608
ELECT 609
ELECT 610
ELECT 611
ELECT 612
ELECT 613
ELECT 614
ELECT 615
ELECT 616
ELECT 617
ELECT 618
ELECT 619
ELECT 620
ELECT 621
ELECT 622
ELECT 623
ELECT 624
ELECT 625
ELECT 626
ELECT 627
ELECT 628

```



	IF (EXPAND) GO TO 90	ELECT	629
C		ELECT	630
C	-----	ELECT	631
C		ELECT	632
	N = 0	ELECT	633
	DO 12 I = 1,NPTS	ELECT	634
	IF (EOVERN(I).EQ.-1.0) GO TO 12	ELECT	635
	N = N+1	ELECT	636
	EOVERN(N) = EOVERN(I)	ELECT	637
	12 CONTINUE	ELECT	638
C		ELECT	639
C	IF NO VALUES OF E/N (UNITS OF 1.0E-16 VOLT CM2) WERE SPECIFIED	ELECT	640
C	BY INPUT, DEFAULT IS TO THE TABLE VALUES *EN* DEFINED ABOVE --	ELECT	641
C		ELECT	642
	IF (N.NE.0) GO TO 25	ELECT	643
	DO 26 I = 1,NPTS	ELECT	644
	IF (EN(I).EQ.0.) GO TO 25	ELECT	645
	N = I	ELECT	646
	26 EOVERN(I) = EN(I)	ELECT	647
	25 NPTS = N	ELECT	648
C		ELECT	649
	NN = 0	ELECT	650
	15 IF (NN.EQ.NPTS) GO TO 60	ELECT	651
	NN = NN+1	ELECT	652
	EVCM = EOVERN(NN)*1.0E-16*NMOL	ELECT	653
	90 CONVRGE = .FALSE.	ELECT	654
C		ELECT	655
C	-----	ELECT	656
C		ELECT	657
	CALL BOLTZ (MGRID=1, MESH, NK, NAME, F1, NGAS, NMOL, TMOL, ITMAX,	ELECT	658
	1 TMAX, EPS, KAPT, TODAY, OUT, EVCM, NE, PROCESS, U, N1, N2, NEL,	ELECT	659
	2 S, BEAM, CREATE, EV, Q, QMOM, F, G, A, B, VSIG, POWER, PCOLL,	ELECT	660
	3 DISCH, DEPOSIT, DEDT, ELASTIC, DNEDT, DLNEDT, IONIZE, ATTACH, VD,	ELECT	661
	4 MU, D, EK, AMPS, UBAR, TE, CONVRGE, PERCENT)	ELECT	662
C		ELECT	663
C	-----	ELECT	664
C		ELECT	665
	EXPAND = (.NOT.CONVRGE).OR.(ABS(PERCENT).GT.PCT)	ELECT	666
	EXPAND = EXPAND.AND.(EMAX.LE.25.)	ELECT	667
	EXPAND = .FALSE.	ELECT	668
	IF (EXPAND) GO TO 40	ELECT	669
	IF (.NOT.CONVRGE) GO TO 60	ELECT	670
C		ELECT	671
	PWR = DISCH * DEPOSIT	ELECT	672
	TABLE(NN,1) = EOVERN(NN)	ELECT	673
	TABLE(NN,2) = UBAR	ELECT	674
	TABLE(NN,3) = EK	ELECT	675
	TABLE(NN,4) = TE	ELECT	676
	TABLE(NN,5) = VD	ELECT	677
	TABLE(NN,6) = MU	ELECT	678
	TABLE(NN,7) = D	ELECT	679
	TABLE(NN,8) = PWR/NMOL	ELECT	680
	TABLE(NN,9) = AMPS	ELECT	681
C		ELECT	682
	IF (PWR.EQ.0.) PWR = AMAX1(ABS(ELASTIC)*ABS(PCOLL),ABS(DEDT))	ELECT	683
	P(NN,NKP1) = ELASTIC*100./PWR	ELECT	684
	DO 4 J = 1,NK	ELECT	685

	P(NN,J) = POWER(J)/PWR*100.	ELECT	686
	IF (VSIG(1,J).GT.0.) RATE(NN,1,J) = ALOG10(VSIG(1,J))	ELECT	687
	4 IF (VSIG(2,J).GT.0.) RATE(NN,2,J) = ALOG10(VSIG(2,J))	ELECT	688
C	GO TO 15	ELECT	689
	60 IF (NN.EQ.1) GO TO 50	ELECT	690
	NPTS = NN	ELECT	691
	NG1 = NGAS-1	ELECT	692
	NSPACE = 58 - 6*NGAS	ELECT	693
	ENCODE (150,230,FORM) NSPACE, NG1, NG1	ELECT	694
C	-----	ELECT	695
C		ELECT	696
C	TABULAR OUTPUT OF PLASMA PARAMETERS FOR THE SPECIFIED GAS MIXTURE	ELECT	697
C	FOR SEVERAL VALUES OF E/N --	ELECT	698
C	IF (.NOT.OUT(8)) GO TO 2	ELECT	699
	L = 22-NPTS	ELECT	700
	IF (L.GT.9) L = 9	ELECT	701
	ENCODE (10,100,SKIP) L	ELECT	702
	WRITE (6,SKIP)	ELECT	703
	WRITE (6,120)	ELECT	704
	IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL	ELECT	705
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =	ELECT	706
	1,NGAS), TMOL	ELECT	707
	WRITE (6,130)	ELECT	708
	DO 52 N = 1,NPTS	ELECT	709
	52 WRITE (6,140) (TABLE(N,I), I = 1,9)	ELECT	710
	WRITE (6,150) EMAX, MESH, DE	ELECT	711
C		ELECT	712
C	TABULAR OUTPUT OF FRACTIONAL POWER PARTITIONING FOR EACH OF THE	ELECT	713
C	ELASTIC AND INELASTIC COLLISION PROCESSES AS A FUNCTION OF E/N --	ELECT	714
C		ELECT	715
	M1 = 1	ELECT	716
	14 IF (M1.GT.NKP1) GO TO 2	ELECT	717
	M2 = M1+20	ELECT	718
	IF (NKP1.LT.M2) M2 = NKP1	ELECT	719
	LSKIP = 20-M2+M1	ELECT	720
	IF (LSKIP.LE.0) LSKIP = 1	ELECT	721
	ENCODE (20,310,OY) LSKIP	ELECT	722
	L1 = 1	ELECT	723
	10 IF (L1.GT.NPTS) GO TO 21	ELECT	724
	L2 = L1+5	ELECT	725
	IF (L2.GT.NPTS) L2 = NPTS	ELECT	726
	NSPACE = (5-L2+L1)*15/2 + 5	ELECT	727
	L21 = L2-L1	ELECT	728
	NM2 = NSPACE-2	ELECT	729
	NDASH = 54 + 15*L21	ELECT	730
	ENCODE (100,280,LABEL) NSPACE, L21, NSPACE	ELECT	731
	ENCODE (20,290,YO) NM2, NDASH	ELECT	732
	ENCODE (50,300,KAPT) NSPACE, L21	ELECT	733
	WRITE (6,OY)	ELECT	734
	WRITE (6,270)	ELECT	735
	IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL	ELECT	736
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =	ELECT	737
	1,NGAS), TMOL	ELECT	738
	WRITE (6,LABEL) (EOVERN(L), L = L1,L2)	ELECT	739
		ELECT	740
		ELECT	741
		ELECT	742



	WRITE (6,Y0)	ELECT	743
	DO 19 N = M1,M2	ELECT	744
	19 WRITE (6,KAPT) (PROCESS(L,N), L = 1,4), (P(L,N), L = L1,L2)	ELECT	745
	WRITE (6,Y0)	ELECT	746
	L1 = L2+1	ELECT	747
	GO TO 10	ELECT	748
	21 M1 = M2+1	ELECT	749
	GO TO 14	ELECT	750
C	-----	ELECT	751
C		ELECT	752
C		ELECT	753
C	PLOTS OF PLASMA PARAMETERS FOR THE SPECIFIED GAS MIXTURE AS A	ELECT	754
C	FUNCTION OF E/NTOT --	ELECT	755
	2 IF (NPTS.LT.5) GO TO 50	ELECT	756
	Y0(1) = Y0(2) = DY(1) = DY(2) = 0.	ELECT	757
	IF (.NOT.OUT(9)) GO TO 30	ELECT	758
C		ELECT	759
	WRITE (6,170)	ELECT	760
	IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL	ELECT	761
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =	ELECT	762
	1,NGAS), TMOL	ELECT	763
	CALL PLOT (21, NPTS, 1, TABLE(1,2), Y0, DY, EOVERN, 0., 0.,	ELECT	764
	1 .TRUE., .TRUE., .TRUE., .TRUE., HEAD(2), 2, 0)	ELECT	765
	WRITE (6,200) TODAY	ELECT	766
C		ELECT	767
	WRITE (6,180)	ELECT	768
	IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL	ELECT	769
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =	ELECT	770
	1,NGAS), TMOL	ELECT	771
	CALL PLOT (21, NPTS, 1, TABLE(1,4), Y0, DY, EOVERN, 0., 0.,	ELECT	772
	1 .TRUE., .TRUE., .FALSE., .TRUE., .TRUE., HEAD(4), 2, 0)	ELECT	773
	WRITE (6,200) TODAY	ELECT	774
C		ELECT	775
	WRITE (6,190)	ELECT	776
	IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL	ELECT	777
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =	ELECT	778
	1,NGAS), TMOL	ELECT	779
	CALL PLOT (21, NPTS, 1, TABLE(1,6), Y0, DY, EOVERN, 0., 0.,	ELECT	780
	1 .TRUE., .TRUE., .FALSE., .TRUE., .TRUE., HEAD(6), 2, 0)	ELECT	781
	WRITE (6,200) TODAY	ELECT	782
C		ELECT	783
	WRITE (6,210)	ELECT	784
	IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL	ELECT	785
	IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =	ELECT	786
	1,NGAS), TMOL	ELECT	787
	CALL PLOT (21, NPTS, 1, TABLE(1,8), Y0, DY, EOVERN, 0., 0.,	ELECT	788
	1 .TRUE., .TRUE., .FALSE., .TRUE., .TRUE., HEAD(8), 2, 0)	ELECT	789
	WRITE (6,200) TODAY	ELECT	790
C		ELECT	791
C		ELECT	792
C	PLOTS OF ENDOTHERMIC ELECTRON COLLISION RATES <VSIG> (CM3/SEC) FOR	ELECT	793
C	ALL OF THE INELASTIC COLLISION PROCESSES, AS A FUNCTION OF E/NTOT	ELECT	794
	30 IF (.NOT.OUT(10)) GO TO 50	ELECT	795
	DO 16 J = 1,NK	ELECT	796
	NPLT = 0	ELECT	797
	IF (VSIG(1,J).GT.0.) NPLT = 1	ELECT	798
		ELECT	799



IF (NPLOT.EQ.0) GO TO 16	ELECT	800
IF (NEL(J).EQ.0) NPLOT = NPLOT+1	ELECT	801
ENCODE (20,160,KAPT) J, J	ELECT	802
WRITE (6,240) J, (PROCESS(I,J), I=1,4)	ELECT	803
IF (NGAS.EV.1) WRITE (6,220) NAME(1), TMOL	ELECT	804
IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =	ELECT	805
1 1,NGAS), TMOL	ELECT	806
CALL PLOT (21, NPTS, 1, RATE(1,1,J), 0., 0., EOVERN, 0., 0.,	ELECT	807
1 .TRUE., .TRUE., .TRUE., .TRUE., .TRUE., KAPT, NPLOT, 0)	ELECT	808
WRITE (6,200) TODAY	ELECT	809
16 CONTINUE	ELECT	810
C GO TO 50	ELECT	811
C	ELECT	812
C ----- FORMAT STATEMENTS -----	ELECT	813
C	ELECT	814
100 FORMAT (5H(1H1.,1,4H(/)))	ELECT	815
C	ELECT	816
110 FORMAT (A10,3E10,3)	ELECT	817
C	ELECT	818
111 FORMAT (12*) *A10,5X)	ELECT	819
C	ELECT	820
112 FORMAT (1H1/20X*WARNING ---//25X*ERRORS OCCURRED FOR THE INPUT E-	ELECT	821
1CROSS SECTIONS FOR THE FOLLOWING ELECTRON COLLISION */25X*PROCESSE	ELECT	822
25. IF THE PROGRAM IS EXECUTED, A ZERO CROSS SECTION WILL BE ASSUM	ELECT	823
3ED:*/20X,90(1H-)//(40X,5A10))	ELECT	824
C	ELECT	825
113 FORMAT (*MOMENTUM TRANSFER FOR *,A10)	ELECT	826
C	ELECT	827
114 FORMAT (14,16)	ELECT	828
C	ELECT	829
115 FORMAT (10(1H) *NO E- CROSS SECTION DATA WAS FOUND.*)	ELECT	830
C	ELECT	831
116 FORMAT (10(1H) *SIGMA = 0 IN THE RANGE (0,*F4.1*) EV.*)	ELECT	832
C	ELECT	833
117 FORMAT (10(1H) *ERRORS OCCURRED IN CROSS SECTION DATA.*)	ELECT	834
C	ELECT	835
118 FORMAT (10(1H) *CROSS SECTION DOES NOT SPAN (0,*F4.1*) EV.*)	ELECT	836
C	ELECT	837
119 FORMAT (10(1H) *THE MASS ENTERED FOR THIS SPECIES IS ≤ 0*)	ELECT	838
C	ELECT	839
120 FORMAT (42X*SUMMARY OF PLASMA PARAMETERS AS A FUNCTION OF E/NTOT*)	ELECT	840
C	ELECT	841
130 FORMAT (/14X,*E/NTOT*,7X,*UBAR*,3X,*EK = D/MU*,5X,*TE*,10X,*VD*,	ELECT	842
1 12X,*MU*,13X,*D*9X*P/NMOL/NE*3X,11HJ/NE = E*VD/11X,*E-16 V CM2)*	ELECT	843
24X*(EV)*,5X*(EV)*,6X*(DEG K)*,5X*(CM/S)*,7X*(CM2/V/S)*,6X,	ELECT	844
3*(CM2/S)*,4X*(WCM3/ELMOL)*,1X*(AMP CM/ELECT)*,10X,116(1H-))	ELECT	845
C	ELECT	846
140 FORMAT (/10X,F10,3,F11,3,F10,3,F11,0,1PSE14,3)	ELECT	847
C	ELECT	848
150 FORMAT (/10X,116(1H-)//36X*(THE ELECTRON ENERGY RANGE (0,*F5.1*) E	ELECT	849
1V WAS DIVIDED INTO *,14/36X* *INTERVALS, GIVING AN ENERGY RESOLU	ELECT	850
2TION OF DU =*,1PE9.2,* EV.)*)	ELECT	851
C	ELECT	852
160 FORMAT (6HVSIGF(,12,2H) ,6HVSIGR(,12,2H) )	ELECT	853
C	ELECT	854
170 FORMAT (1H1/44X*AVERAGE AND CHARACTERISTIC ELECTRON ENERGY (EV)*)	ELECT	855
	ELECT	856

C	180 FORMAT (1H142X*EFFECTIVE ELECTRON TEMPERATURE AND DRIFT VELOCITY*)	ELECT	857
		ELECT	858
C	190 FORMAT (1H1/46X*ELECTRON MOBILITY AND DIFFUSION COEFFICIENT*)	ELECT	859
		ELECT	860
C	200 FORMAT (/56X*E/NTOT (1.0 E-16 VOLT CM2)*18X*DR. WILLIAM B. LACINA, 1*A11/100X*NORTHROP RESEARCH AND TECHNOLOGY*)	ELECT	861
		ELECT	862
C	210 FORMAT (1H142X*TOTAL ELECTRICAL POWER AND CURRENT DENSITY PER NE*)	ELECT	863
		ELECT	864
C	220 FORMAT (55X,*PURE *,A3,* TMOL =*,F5.0,* DEG K*/)	ELECT	865
		ELECT	866
C	230 FORMAT (1H(,I2,4HXA3,,I2,20H(*/*A3),* = *2PF6.2,,I2,38H(* /*F6.2), 1*, TMOL =*,0PF5.0* DEG K*/))	ELECT	867
		ELECT	868
C	240 FORMAT (1H1,33X,*LOG PLOT OF ELECTRON COLLISION RATES <VSIG> (CM3/ 1SEC) FOR REACTION (*I2*)*/55X,4A10/)	ELECT	869
		ELECT	870
C	250 FORMAT (8A10)	ELECT	871
		ELECT	872
C	260 FORMAT (80A1)	ELECT	873
		ELECT	874
C	270 FORMAT (18X*SUMMARY OF FRACTIONAL % POWER PARTITION FOR ALL ELECTR ION COLLISION PROCESSES AS A FUNCTION OF E/NTOT*)	ELECT	875
		ELECT	876
C	280 FORMAT (*(/*I2*X,4X*REACTION*15X*E/NTOT --> *F7.2,*I1*F15.2/*, I12*X,24X*(1.0E-16 V CM2)*))	ELECT	877
		ELECT	878
C	290 FORMAT (1H(,I2*X,*I3*(1H-)/))	ELECT	879
		ELECT	880
C	300 FORMAT (1H(,I2*X,4A10,F8.3,*I1*F15.3/))	ELECT	881
		ELECT	882
C	310 FORMAT (5H(1H1,,I2*(/))*)	ELECT	883
		ELECT	884
	320 FORMAT (*MOMENTUM TRANSFER COLLISIONS*)	ELECT	885
		ELECT	886
C	330 FORMAT (50X*NO FURTHER WARNING DIAGNOSTICS WILL BE ISSUED.*)	ELECT	887
		ELECT	888
C	340 FORMAT (50X)	ELECT	889
		ELECT	890
C	350 FORMAT (/30X*FATAL ERROR -- MOMENTUM TRANSFER COLLISION FREQUENCY* 1/30X*BECAME ZERO AT SOME POINT; ANALYSIS HAS 1/OM TERMS.*)	ELECT	891
		ELECT	892
C	360 FORMAT (1H1,22X, *NORMALIZED EXTERNAL SOURCE FUNCTION S(U) FOR C 1REATION OF ELECTRONS IN THE ENERGY RANGE (U,U*DU)*/25X*TOTAL ELECT 2RON CREATION RATE = *1PE10.3* CM-3/SEC. POWER DEPOSITION = *E10.3, 3* WATT/CM3*/)	ELECT	893
		ELECT	894
C	370 FORMAT (/62X*ELECTRON ENERGY U (EV)*16X*DR. WILLIAM B. LACINA,*A11 1/100X*NORTHROP RESEARCH AND TECHNOLOGY*)	ELECT	895
		ELECT	896
C	500 FORMAT (7X*ELECTRON KINETICS ANALYSIS*7X)	ELECT	897
		ELECT	898
C	-----	ELECT	899
		ELECT	900
C	99 CALL EXIT	ELECT	901
	END	ELECT	902
		ELECT	903
		ELECT	904
		ELECT	905
		ELECT	906
		ELECT	907
		ELECT	908
		ELECT	909
		ELECT	910
		ELECT	911
		ELECT	912
		ELECT	913